EXHIBIT 10

US006857116B1

(12) **United States Patent**
Dahl et al.

(10) **Patent No.: US 6,857,116 B1**
(45) **Date of Patent: Feb. 15, 2005**

(54) **OPTIMIZATION OF ABUTTED-PIN HIERARCHICAL PHYSICAL DESIGN**

(75) Inventors: **Peter Dahl**, Cupertino, CA (US); **Byron Dickinson**, San Jose, CA (US); **Margie Levine**, Menlo Park, CA (US); **Paul Rodman**, Palo Alto, CA (US)

(73) Assignee: **Reshape, Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/714,722**

(22) Filed: **Nov. 15, 2000**

(51) **Int. Cl.**[7] **G06F 17/50**
(52) **U.S. Cl.** **716/12**; 716/13; 716/14
(58) **Field of Search** 716/2, 4, 7, 8, 716/9, 11, 12, 13, 14

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,890,238	A	12/1989	Klein et al.	364/491
5,187,671	A	2/1993	Cobb	364/490
5,309,370	A	5/1994	Wong	364/490
5,533,148	A	7/1996	Sayah et al.	382/240
5,544,088	A	8/1996	Aubertine et al.	364/489
5,576,969	A	11/1996	Aoki et al.	364/491
5,757,658	A	5/1998	Rodman et al.	364/491
6,243,854	B1	6/2001	Lavin et al.	716/19
6,425,113	B1 *	7/2002	Anderson et al.	716/5
6,618,849	B2	9/2003	Teig et al.	716/12

FOREIGN PATENT DOCUMENTS

WO	WO00/67163	11/2000	G06F/17/50

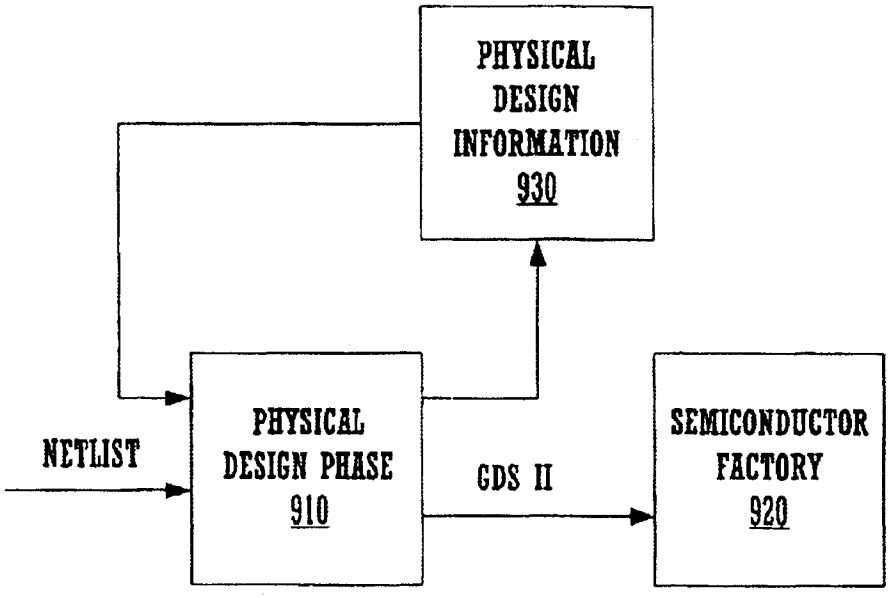
* cited by examiner

Primary Examiner—Thien F Tran
(74) *Attorney, Agent, or Firm*—Wagner, Murabito, & Hao LLP

(57) **ABSTRACT**

An abutted-pin hierarchical physical design process is described. The abutted-pin hierarchical physical design provides solutions to the problems of the traditional hierarchical physical design and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced significantly.

52 Claims, 31 Drawing Sheets



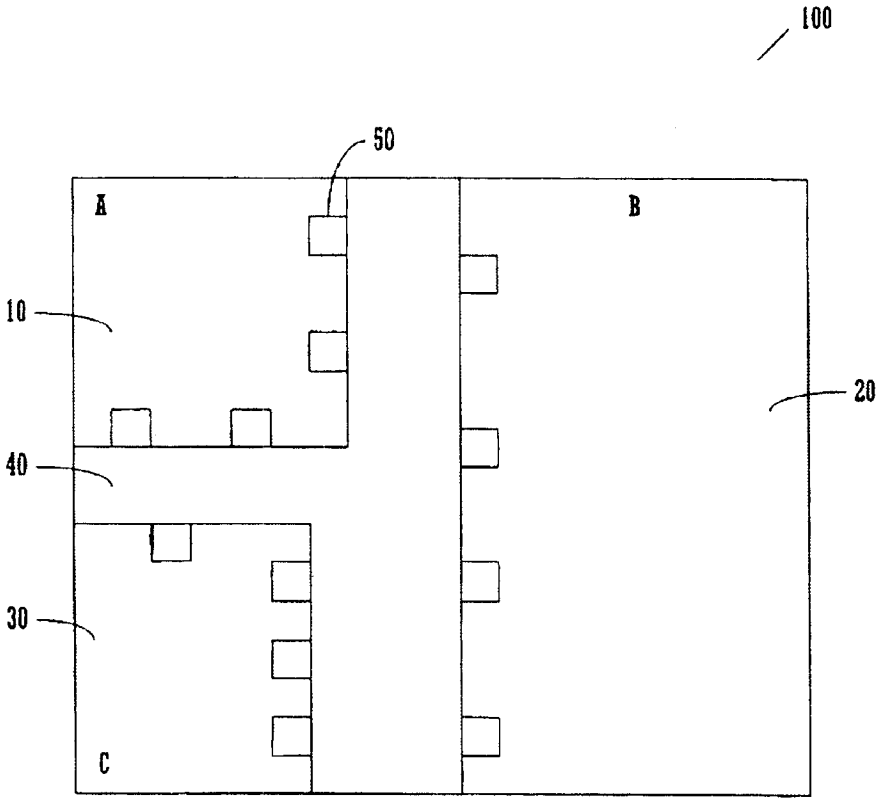


FIGURE 1
(Prior Art)

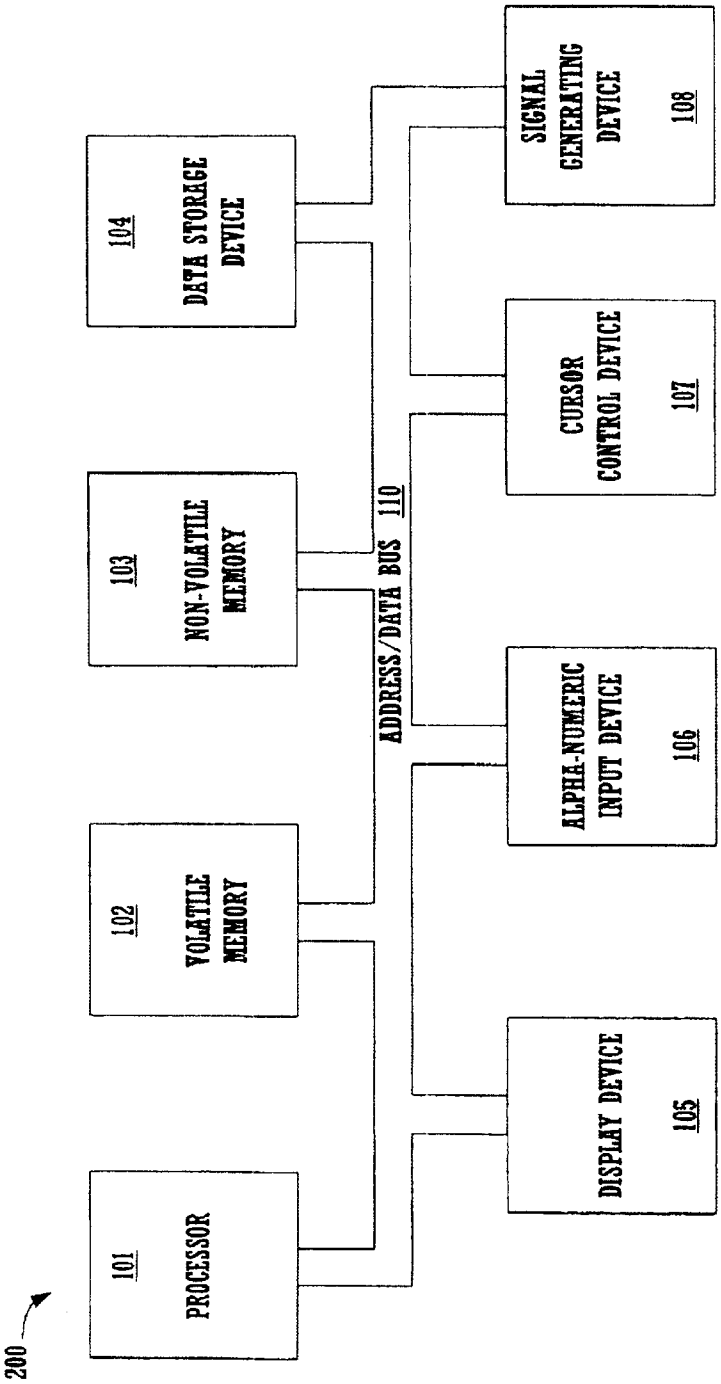

200
101 PROCESSOR
102 VOLATILE MEMORY
103 NON-VOLATILE MEMORY
104 DATA STORAGE DEVICE
ADDRESS/DATA BUS 110
DISPLAY DEVICE 105
ALPHA-NUMERIC INPUT DEVICE 106
CURSOR CONTROL DEVICE 107
SIGNAL GENERATING DEVICE 108

FIGURE 2

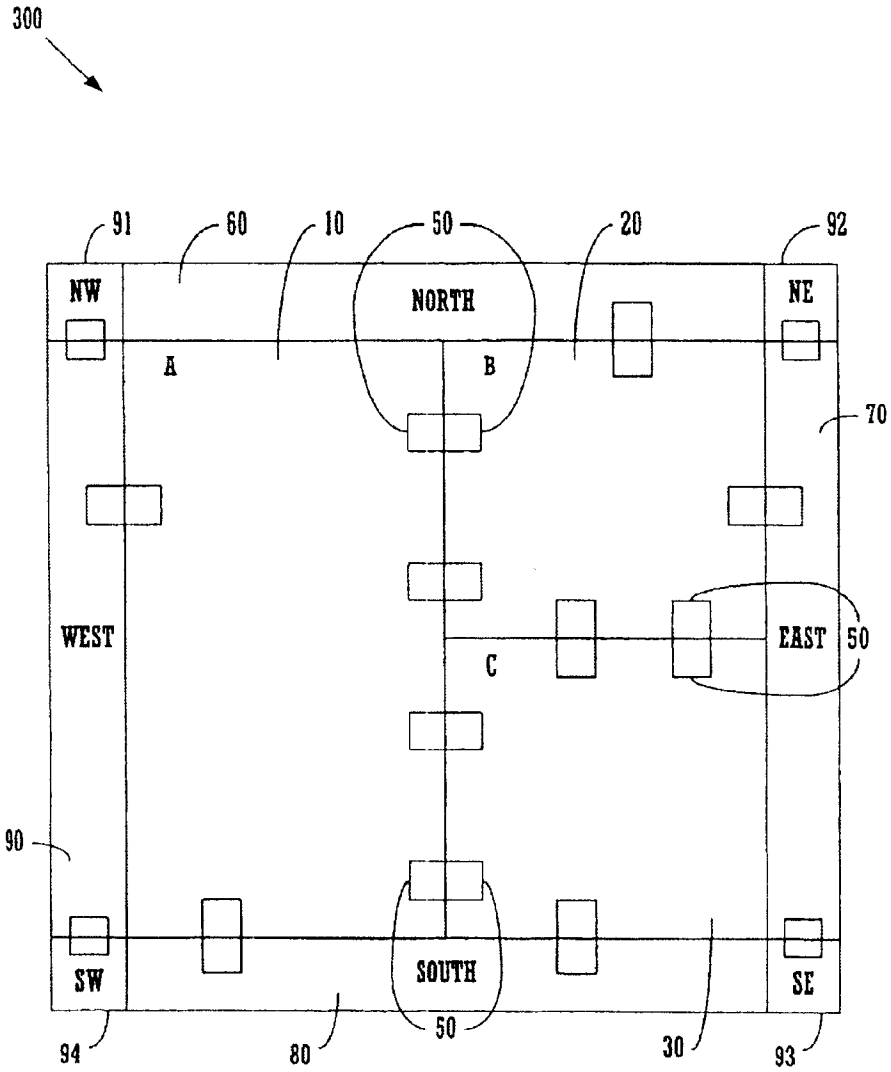

300
91
60
10
50
20
92
NW
NORTH
NE
A
B
70
WEST
C
EAST
50
90
SW
SOUTH
SE
50
94
80
30
93

FIGURE 3

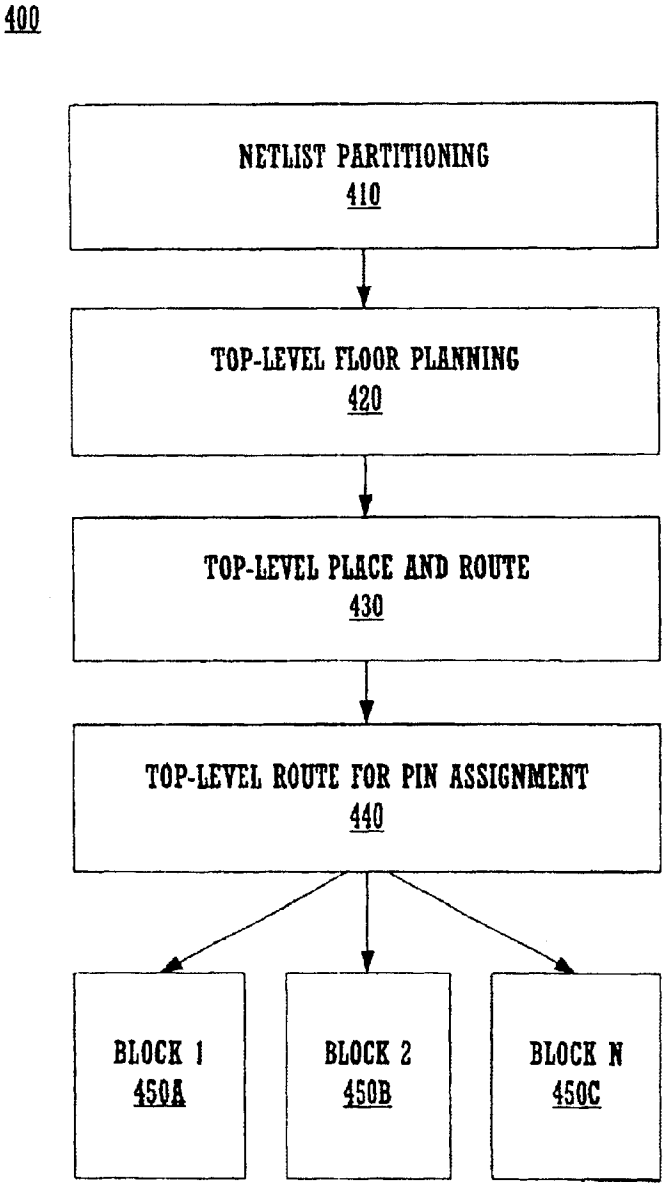

400
NETLIST PARTITIONING
410
TOP-LEVEL FLOOR PLANNING
420
TOP-LEVEL PLACE AND ROUTE
430
TOP-LEVEL ROUTE FOR PIN ASSIGNMENT
440
BLOCK 1
450A
BLOCK 2
450B
BLOCK N
450C

FIGURE 4

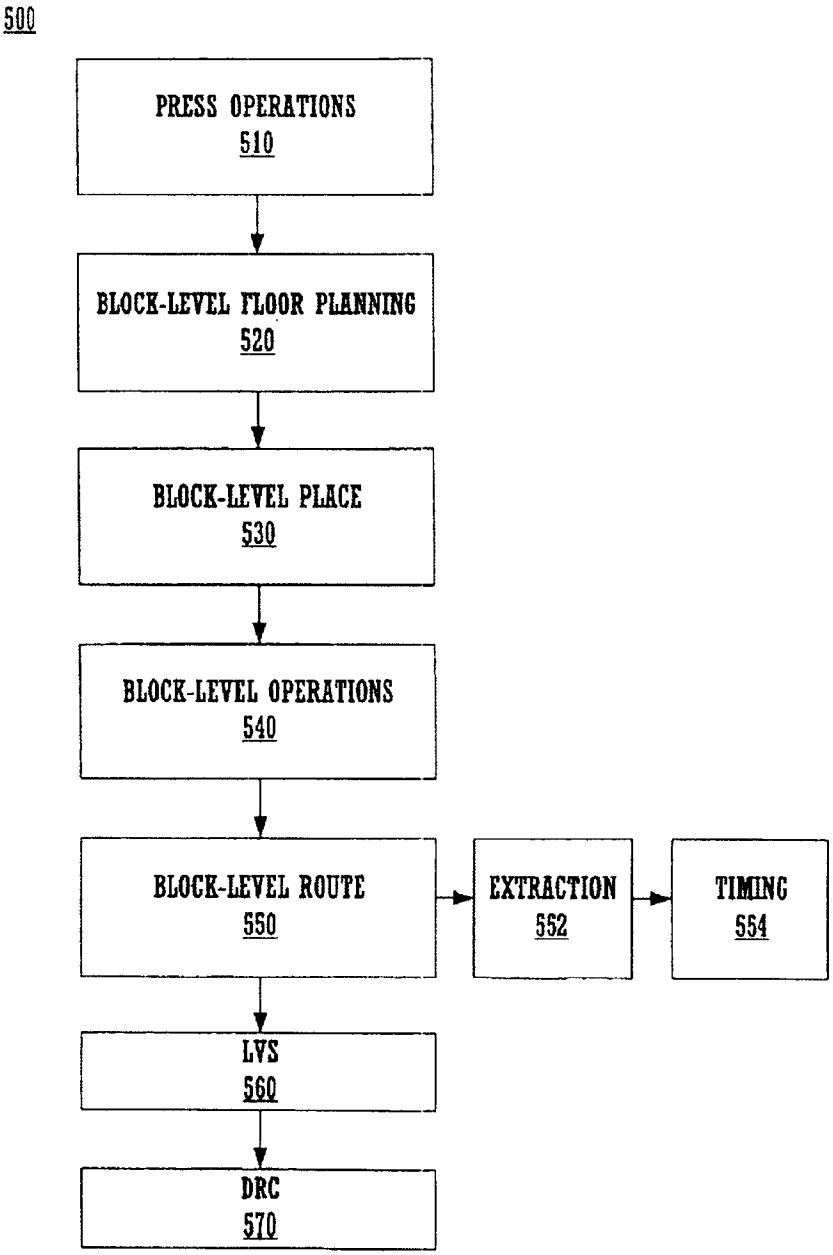

500
PRESS OPERATIONS
510
BLOCK-LEVEL FLOOR PLANNING
520
BLOCK-LEVEL PLACE
530
BLOCK-LEVEL OPERATIONS
540
BLOCK-LEVEL ROUTE
550
EXTRACTION
552
TIMING
554
LVS
560
DRC
570

FIGURE 5

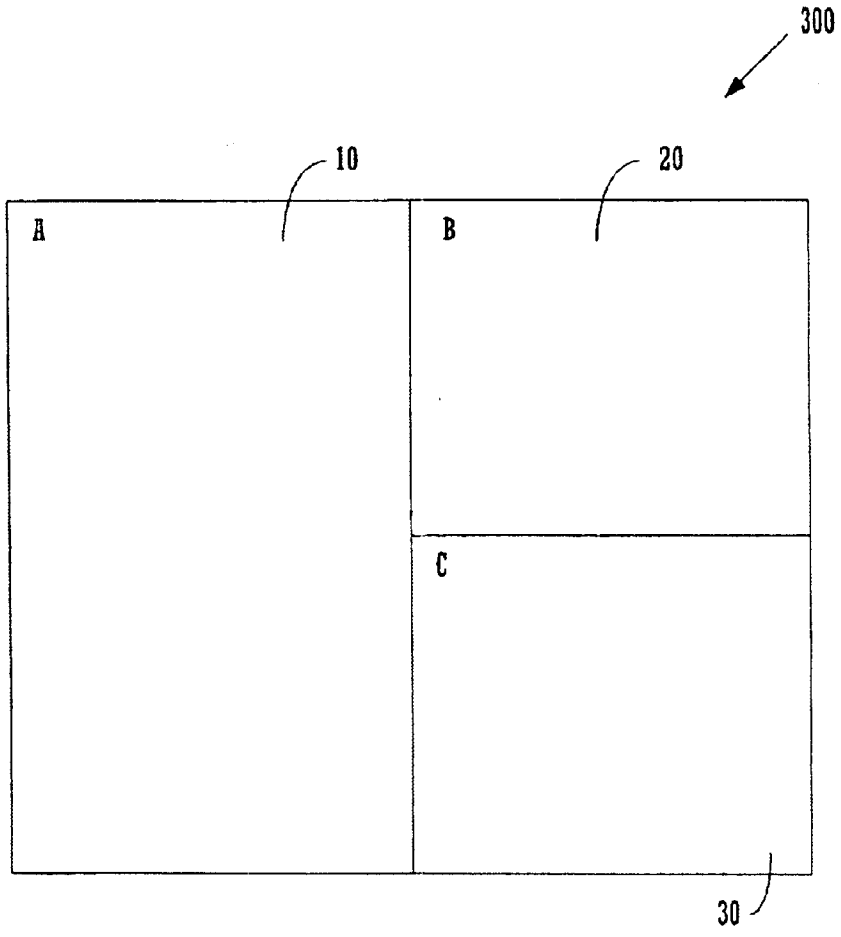


FIGURE 6

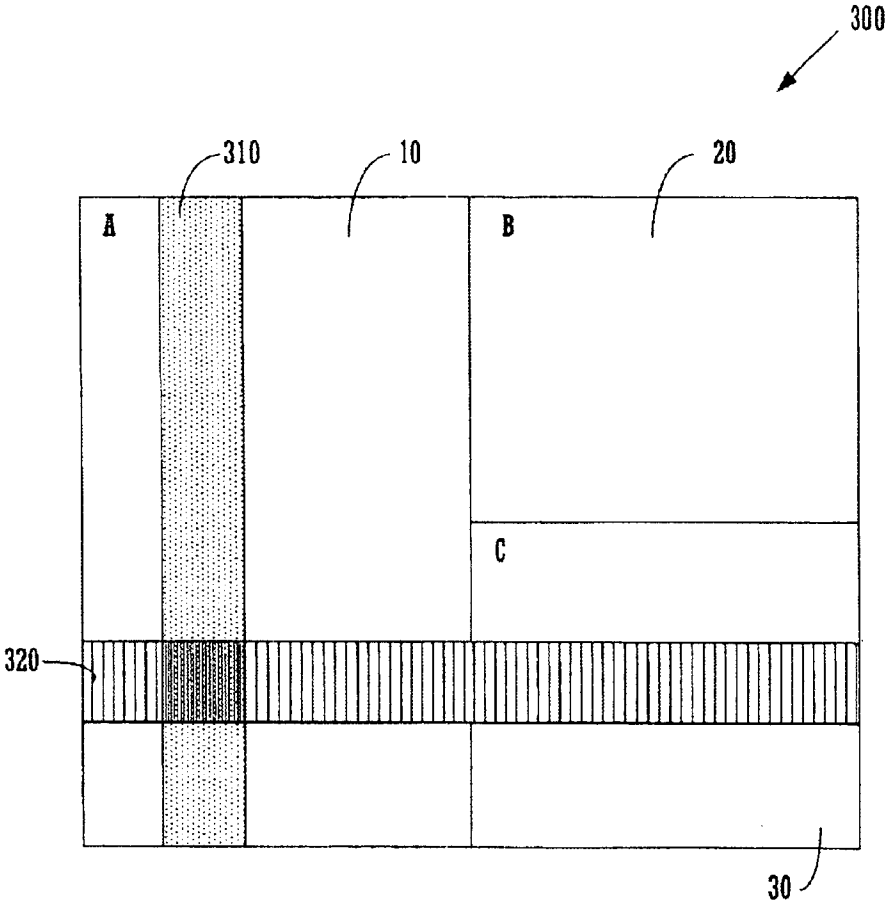


FIGURE 7

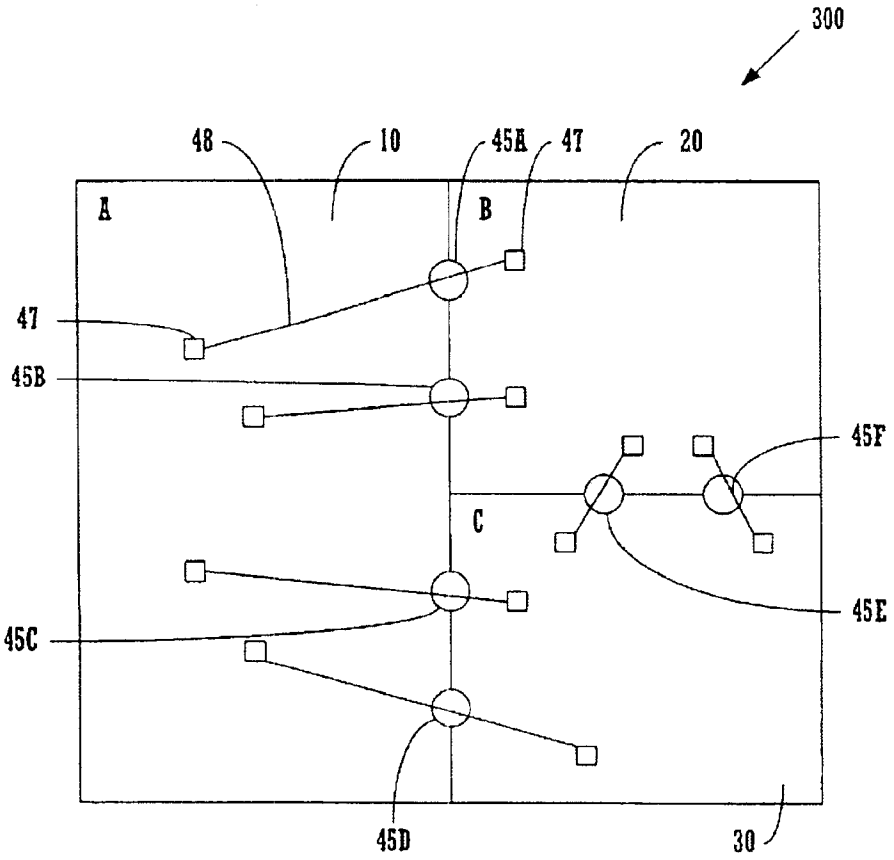


FIGURE 8

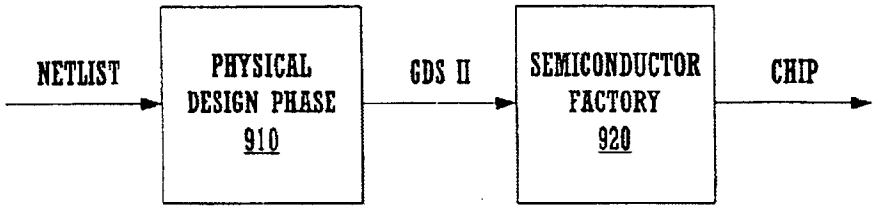


FIGURE 9A
(Prior Art)

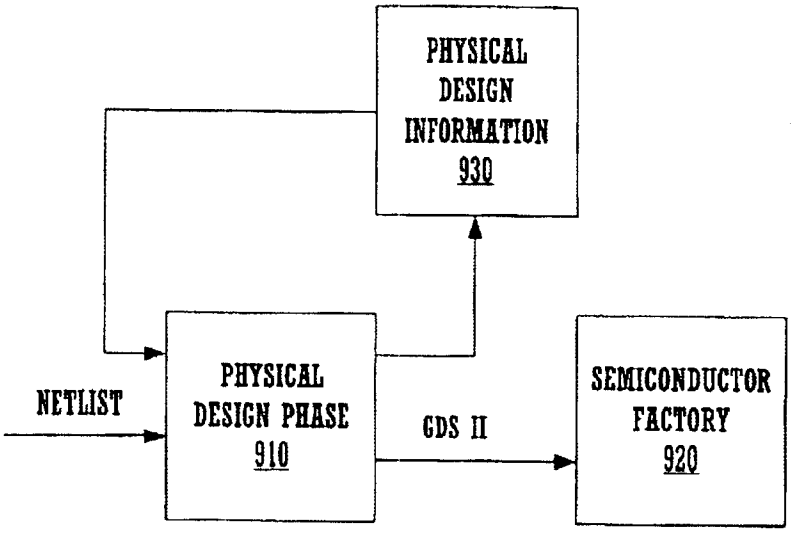

PHYSICAL
DESIGN
INFORMATION
930
NETLIST
PHYSICAL
DESIGN PHASE
910
GDS II
SEMICONDUCTOR
FACTORY
920

FIGURE 9B

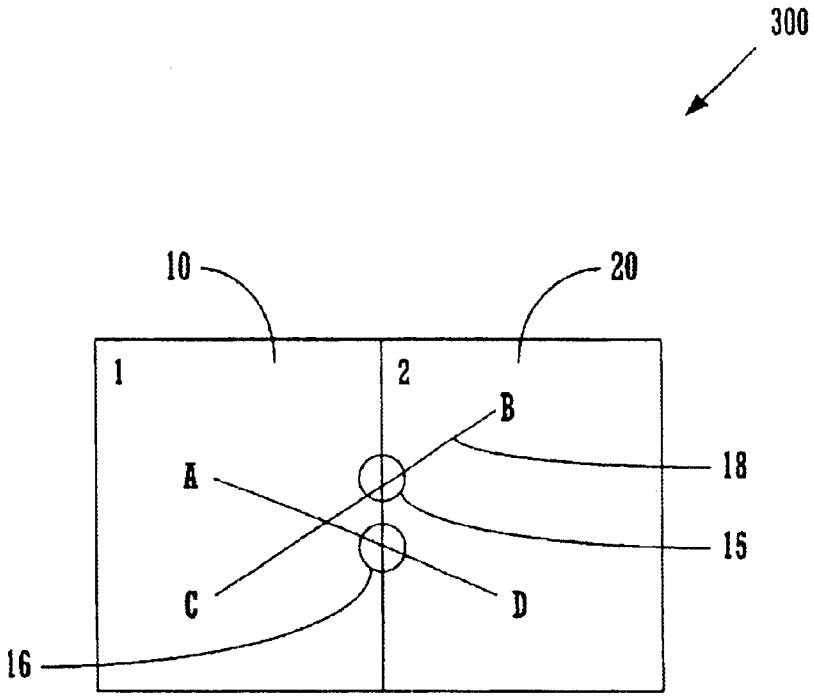


FIGURE 10A

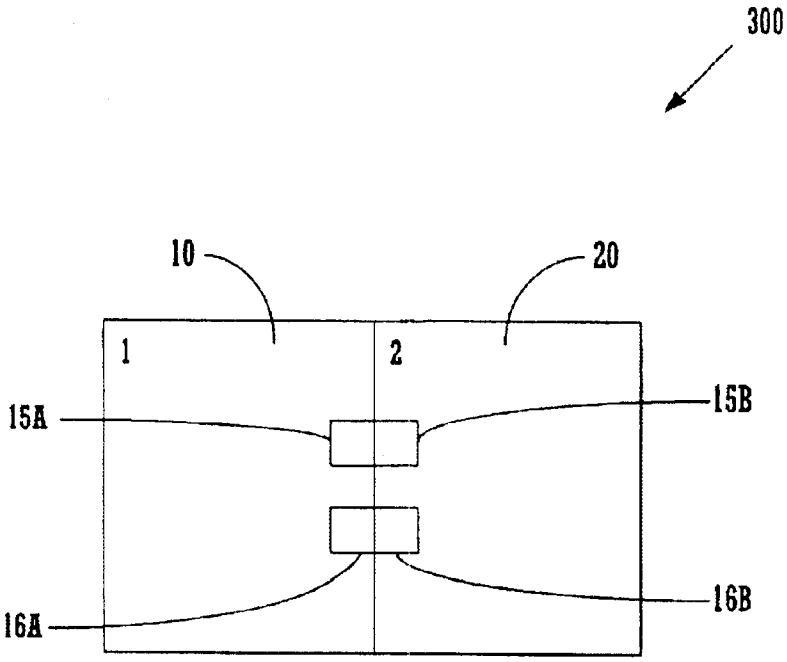


FIGURE 10B

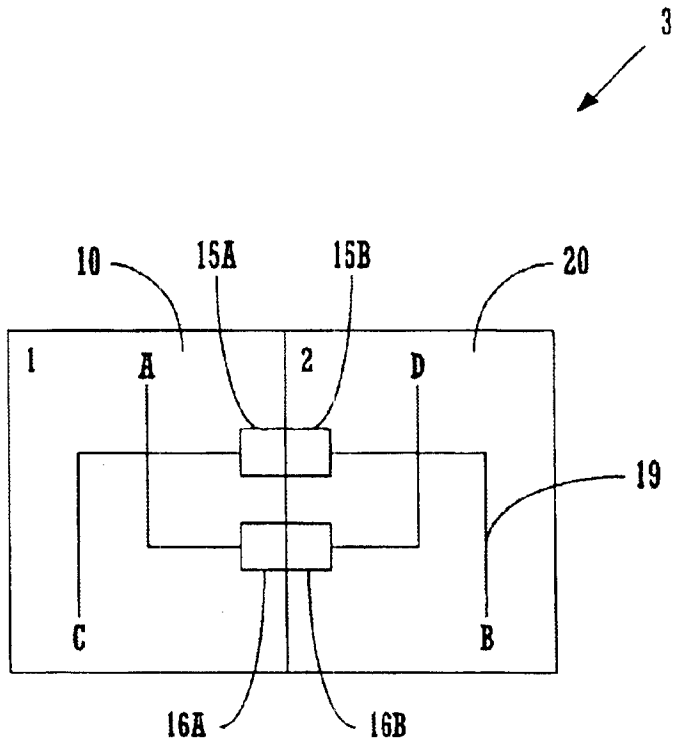


FIGURE 10C

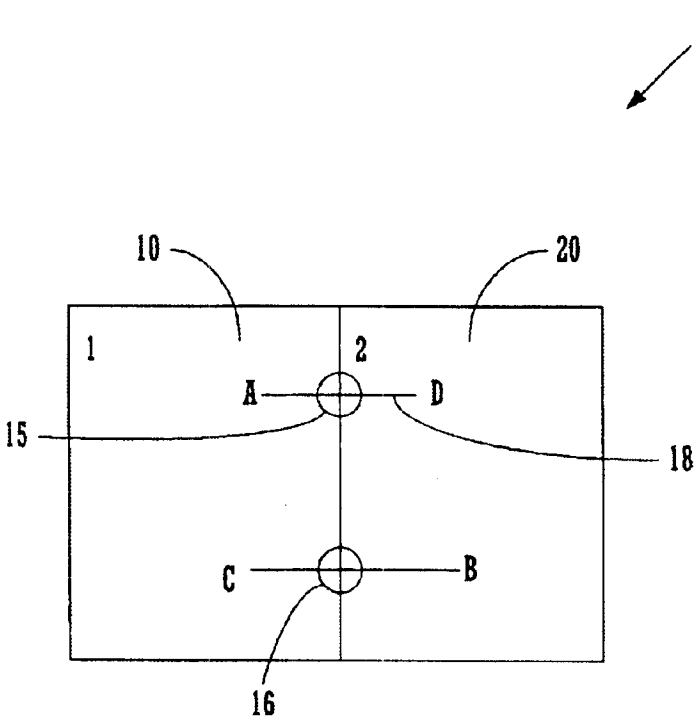


FIGURE 11A

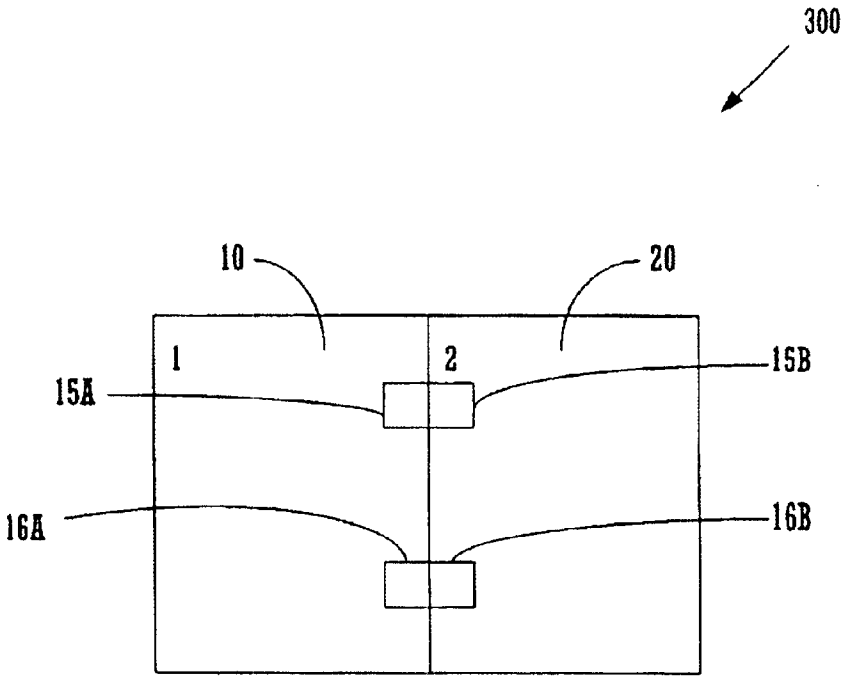


FIGURE 11B

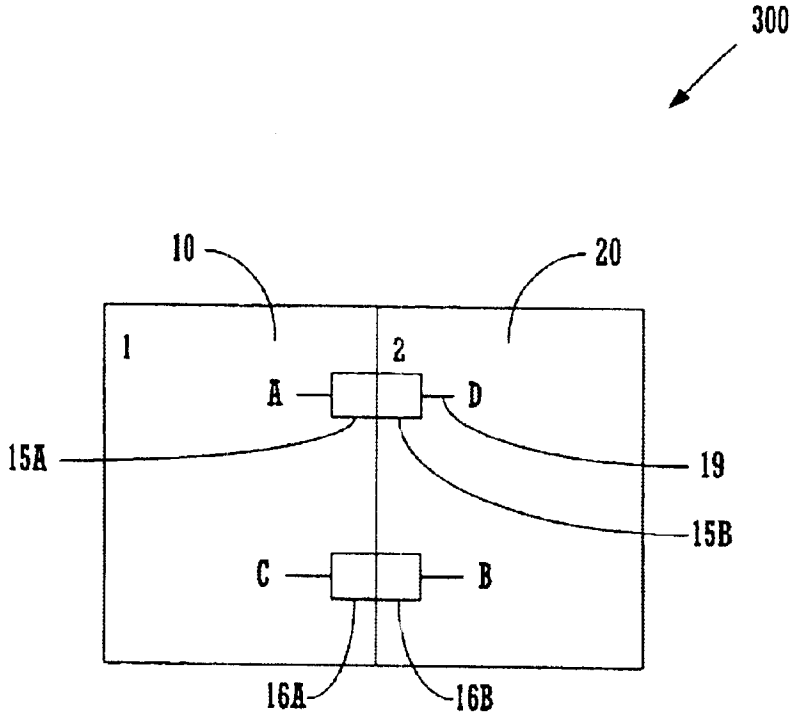


FIGURE 11C

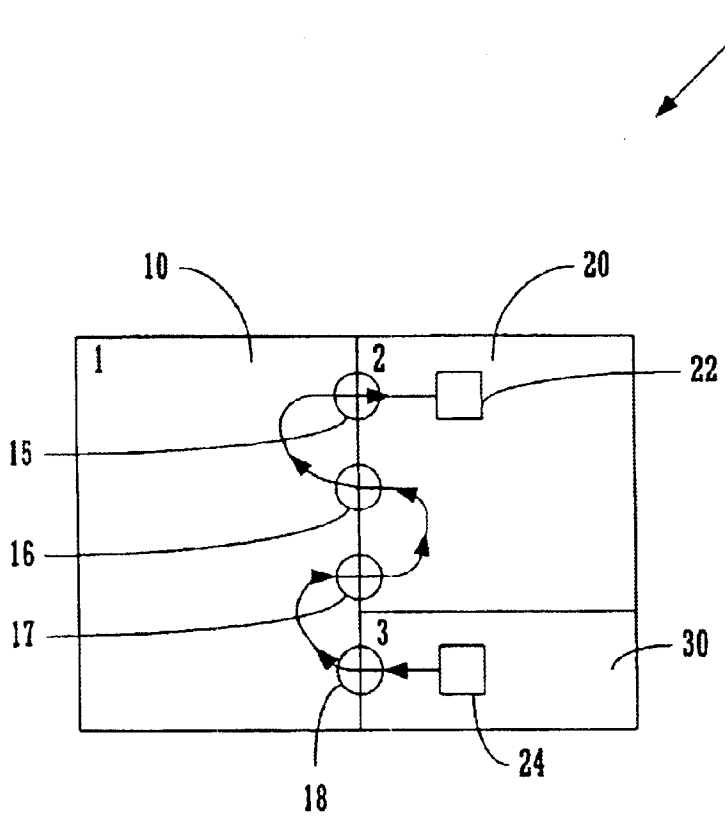


FIGURE 12A

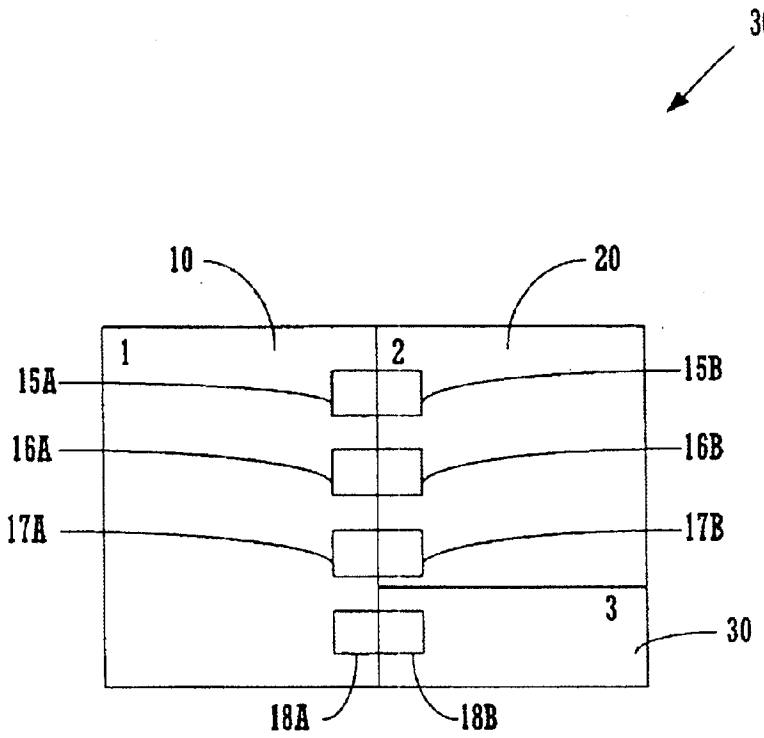


FIGURE 12B

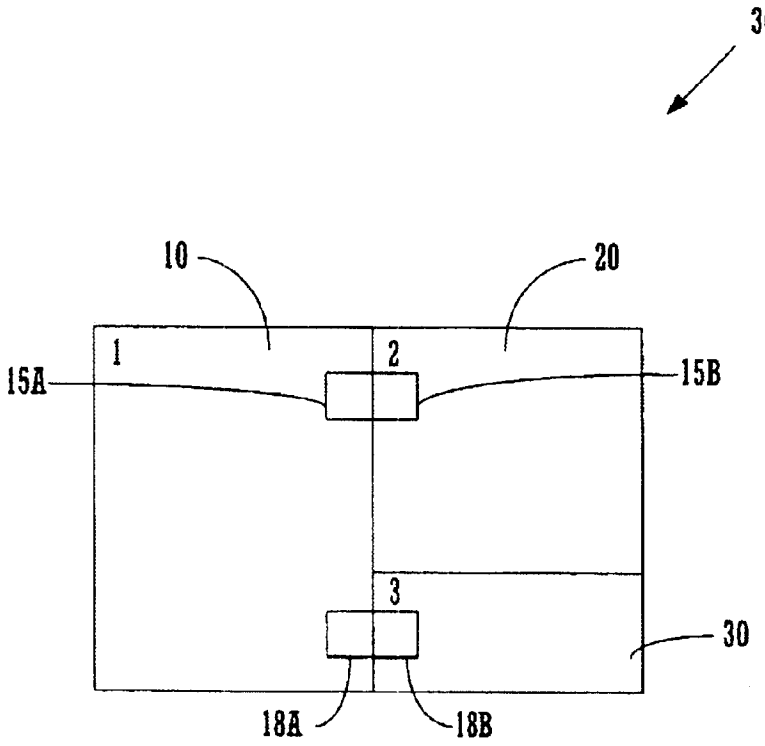


FIGURE 12C

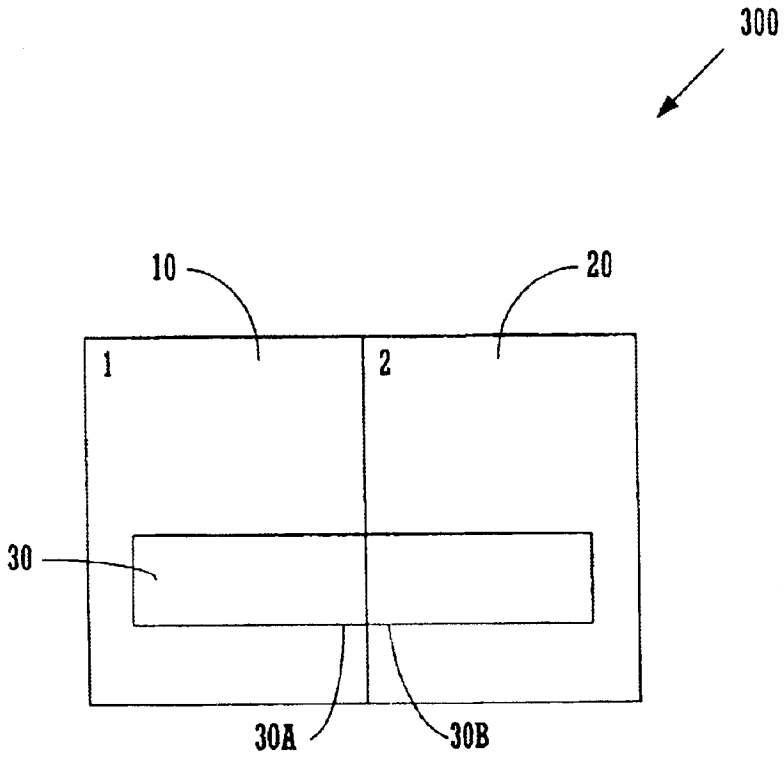


FIGURE 13A

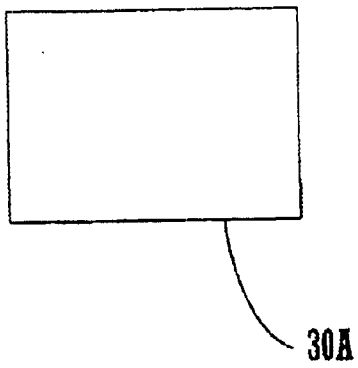


FIGURE 13B

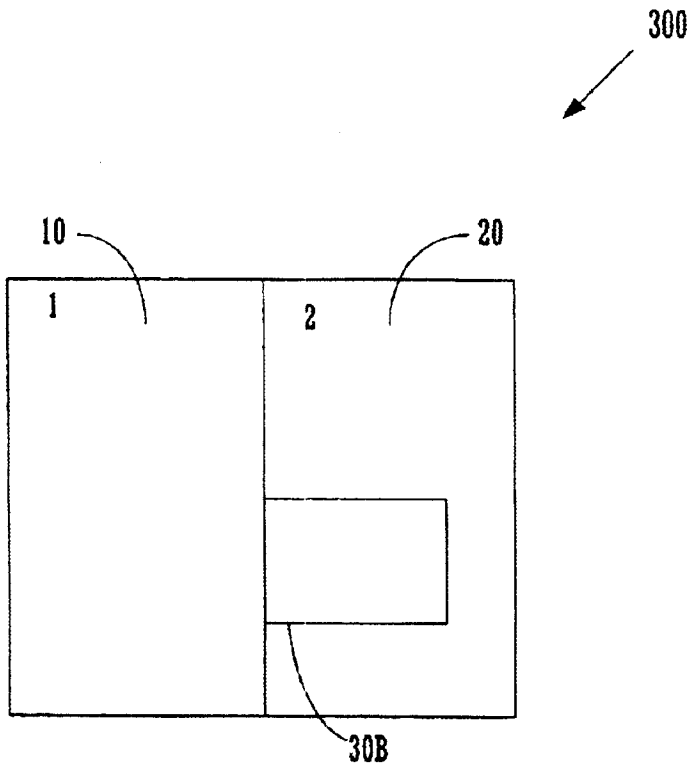


FIGURE 13C

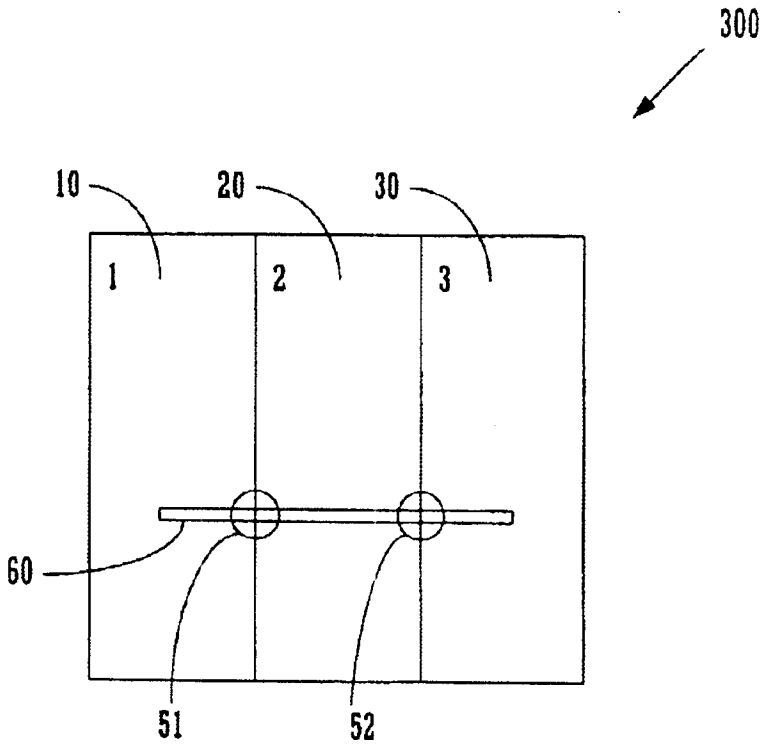


FIGURE 14A

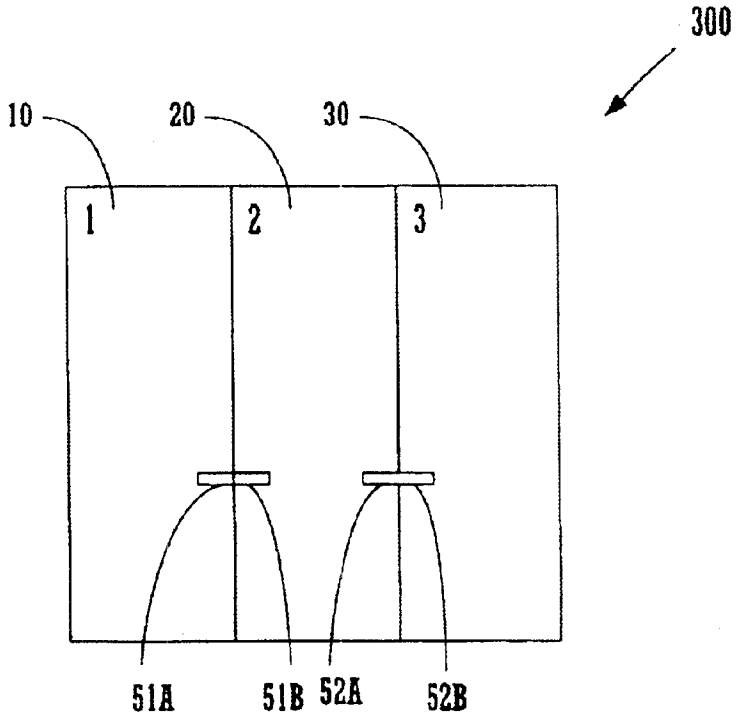


FIGURE 14B

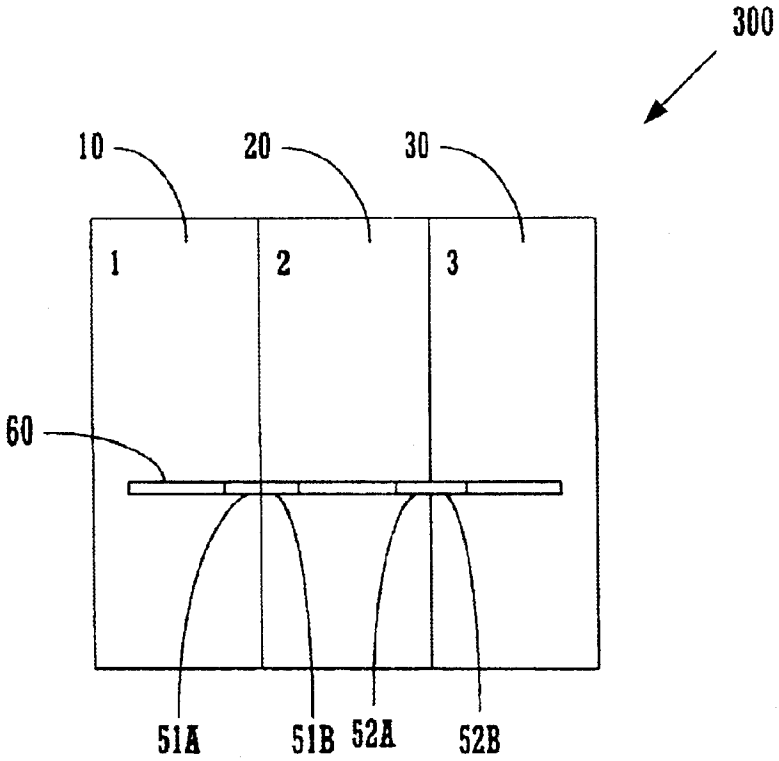


FIGURE 14C

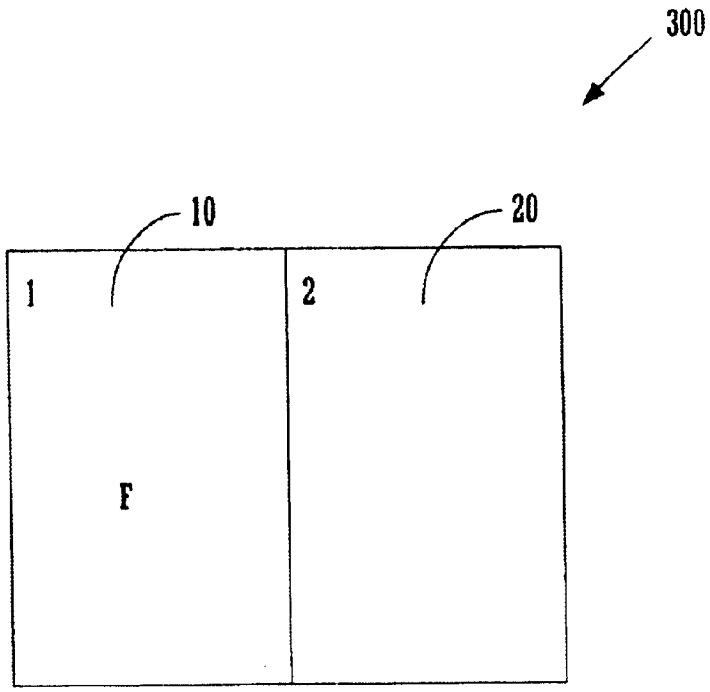


FIGURE 15A

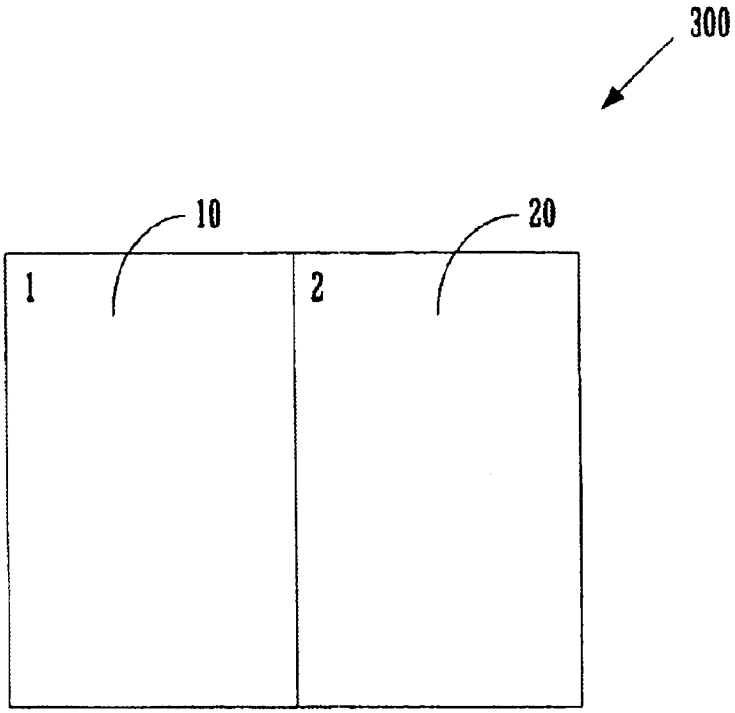


FIGURE 15B

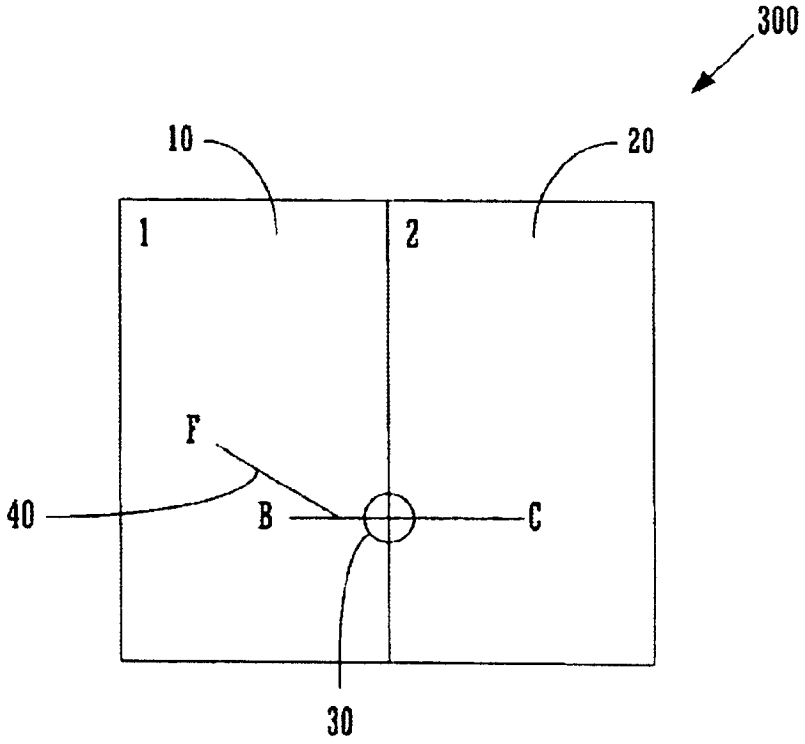


FIGURE 16A

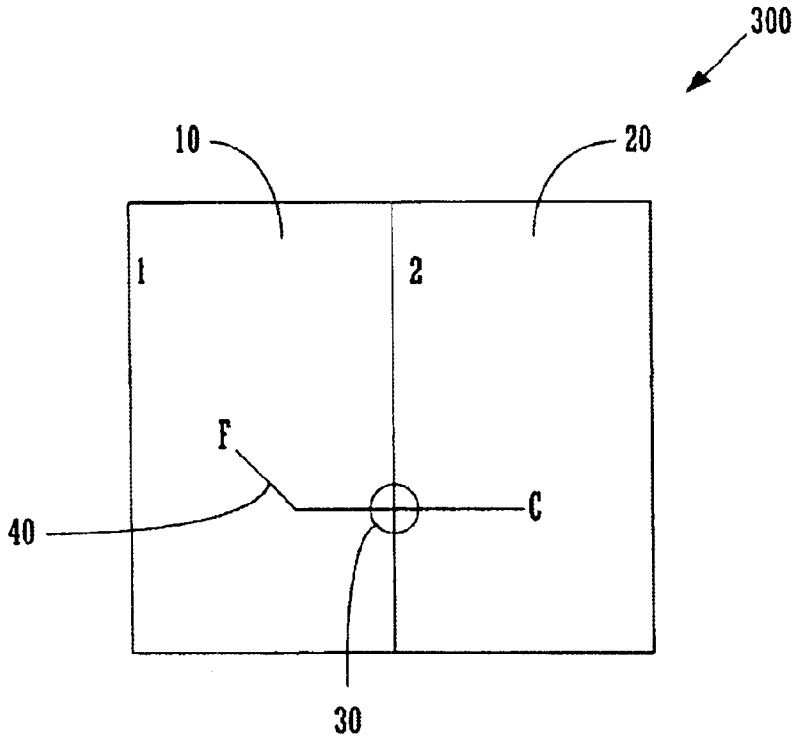


FIGURE 16B

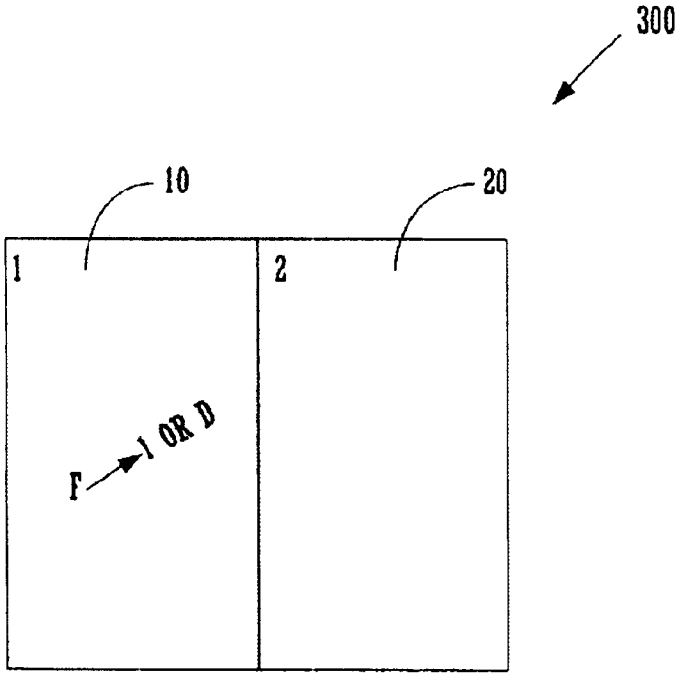


FIGURE 17A

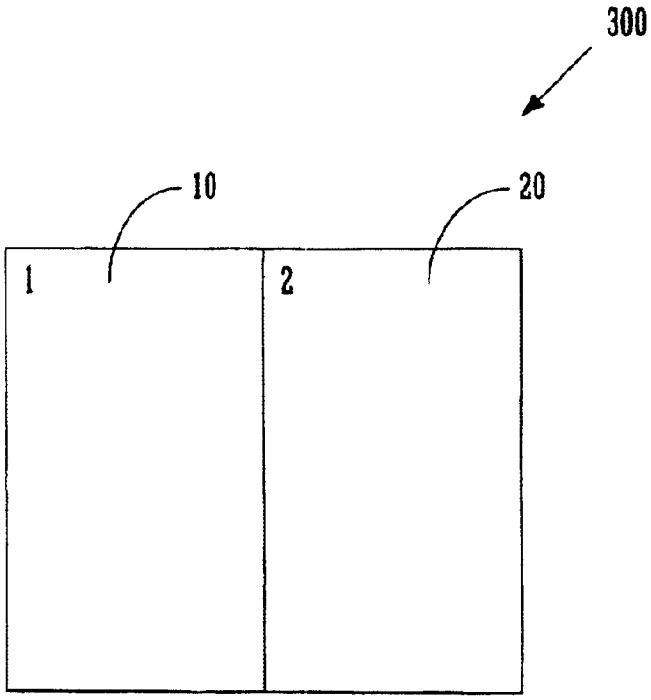


FIGURE 17B

OPTIMIZATION OF ABUTTED-PIN HIERARCHICAL PHYSICAL DESIGN

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to the field of integrated circuit design. More particularly, the present invention relates to the field of software tools for hierarchical physical design.

2. Related Art

The tremendous advances in technology have been fueled by improvements in integrated circuit design. In particular, integrated circuits have become smaller and more complex. Integrated circuit design engineers depend on electronic design automation (EDA) software tools to facilitate the design of integrated circuits.

Typically, the integrated circuit design process begins with a specification which describes the functionality of the integrated circuit and may include a variety of constraints. Then, during a logic design phase, the logical implementation of the integrated circuit is determined. Several operations are performed to obtain a logical representation of the integrated circuit. Generally, EDA software tools use register transfer logic (RTL) to represent the integrated circuit. However, additional EDA software tools may be used.

After completing the logic design phase, the integrated circuit undergoes a physical design phase. Typically, the output of the logic design phase, is a netlist, which is then used in the physical design phase. Here, EDA software tools layout the integrated circuit to obtain a representation of the physical components in the integrated circuit, whereas the representation indicates the manner in which the integrated circuit will be implemented on a semiconductor chip. A variety of operations are performed on the layout of the integrated circuit.

At the end of the physical design phase, the representation of the semiconductor chip (in which the integrated circuit is implemented) is sent to a semiconductor manufacturing plant.

Typically, in the physical design phase, EDA software tools implement a flat physical design. For example, the components (standard cells, macrocells, etc.) of the integrated circuit are placed during a placement operation and are routed during a routing operation. However, as the integrated circuit becomes more complex, the EDA software tools struggle to perform the placement operation and the routing operation. In particular, the performance of the EDA software tools degrades since the EDA software tools have to manipulate very large files during the placement operation and the routing operation. Moreover, as the complexity of the integrated circuit increases, the time necessary to complete the physical design phase increases significantly.

Traditional hierarchical physical design has emerged as an alternative to the flat physical design. FIG. **1** illustrates the traditional hierarchical physical design **100**. Here, the components of the integrated circuit are partitioned into a plurality of blocks **10–30**. Each block **10–30** includes a plurality of pins **50**, whereas each pin **50** represents a location where a signal can enter the block **10–30** or a location where a signal can exit the block **10–30**. As illustrated in FIG. **1**, the traditional hierarchical physical design **100** includes a channel **40**. The channel **40** provides space in order to connect the pins **50** of the blocks **10–30** to one another via metal (not shown) or any other wiring material. The traditional hierarchical physical design **100** enables the placement operation and the routing operation (as well as other operations) for the blocks **10–30** to be performed in parallel with EDA software tools, reducing the time period of the physical design phase. Moreover, the performance of the EDA software tools is improved because the file for each block **10–30** is much smaller than the file for the entire integrated circuit of the flat physical design. More importantly, the EDA software tools are better suited to optimize each block **10–30** than to optimize the entire integrated circuit of the flat physical design. However, the traditional hierarchical physical design **100** generates wasted space in the channel **40** and generates wiring problems in the channel **40**, such as congestion and crosstalk. Moreover, the traditional hierarchical physical design **100** places and routes components at a top-level (shown in FIG. **1**) and a block-level (within each block **10–30**), causing inefficiencies and causing problems with EDA software tools which are configured to operate with flat physical designs.

SUMMARY OF THE INVENTION

An abutted-pin hierarchical physical design process is described. The abutted-pin hierarchical physical design provides solutions to the problems of the traditional hierarchical physical design and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced significantly.

In the integrated circuit design flow according to an embodiment of the present invention, the physical design phase receives the netlist from the logic design phase. In addition, the physical design phase receives physical design information, whereas the physical design information can be any information about a prior integrated circuit that has undergone the physical design phase. In an embodiment, the physical design information is stored in a database.

In an embodiment of the present invention, the integrated circuit design flow of the present invention is utilized to optimize pin assignment. In an embodiment of the present invention, excess pins formed along a boundary between two blocks are removed.

In an embodiment of the present invention, a software tool that performs a "press" operation preserves the properties associated with a segment of a top-level shape despite the shape operation (e.g., AND) being performed with the block and the top-level shape to obtain the segment.

If the top-level object has the press property, the top-level object retains its location when the top-level object is "pressed" into a block. If the top-level object does not have the press property, the top-level object generally does not retain its location when the top-level object is "pressed" into the block.

If in the top-level netlist, the instantiation of a block includes a port that is unused, (thus, not needed for the top-level routing for pin assignment), a software tool removes the port from the top-level netlist, but the block-level netlist of the block remains unchanged.

Some software tools are not able to represent the relationship that more than one port is coupled to a pin. Hence, a software tool removes one of the ports from the netlist based on some criteria, such as whether a port is an input port or an output port.

If in the top-level netlist, the instantiation of the block includes a port that is tied to either the power line (**1**) or the

ground line (**0**) rather to a port of another block, a software tool removes the port from the top-level netlist to avoid routing the port at the top-level. Moreover, the software tool ties the port to either the power line (**1**) or the ground line (**0**) in the block-level netlist of the block.

In an embodiment, a software tool performs an unwinding operation which adds to the block-level netlist---of bonding pad blocks---the ports (which were removed earlier by the software tool) that couple to the top-level inputs and to the top-level outputs. Thus, the netlist modified by the physical design phase (e.g., repeater and buffers are added to the netlist) can be compared with the netlist originally received from the logic design phase. In particular, formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification can be performed by software tools.

In an embodiment, each block-level netlist is partitioned into a first netlist and a second netlist. The second netlist and its associated extraction file of each block and the top-level netlist and its associated extraction file are utilized by software tools to perform the timing analysis. This timing analysis can be performed significantly faster than the case where the block-level netlist is not partitioned into the first netlist and the second netlist. In an embodiment, the timing graph resulting from the timing analysis can be analyzed to extract timing constraints (relating to the delay that can be generated by a block) for each block. Hence, if a block is optimized to meet its extracted timing constrains, the block is more likely to meet its timing parameter when the block interacts with the other blocks in the integrated circuit.

These and other advantages of the present invention will no doubt become apparent to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the present invention.

FIG. **1** illustrates the traditional hierarchical physical design **100**.

FIG. **2** illustrates an exemplary computer system **200** on which embodiments of the present invention may be practiced.

FIG. **3** illustrates an integrated circuit **300** generated with software tools according to an embodiment of the abutted-pin hierarchical physical design process of the present invention.

FIG. **4** illustrates the abutted-pin hierarchical physical design process **400** according to an embodiment of the present invention.

FIG. **5** illustrates the abutted-pin hierarchical physical design process **500** as performed at the block-level in a particular block (**450**A–**450**C of FIG. **4**) after step **440** of FIG. **4**.

FIG. **6** illustrates the layout of the blocks **10**–**30** is established.

FIG. **7** illustrates a clock wire **320** and a power wire **310** of the top-level.

FIG. **8** illustrates a top-level route for obtaining the pin assignments for each block **10**–**30**.

FIG. **9**A illustrates the integrated circuit design flow of the prior art.

FIG. **9**B illustrates the integrated circuit design flow according to an embodiment of the present invention.

FIG. **10**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the prior art (FIG. **9**A), showing the top-level routing for pin assignment.

FIG. **10**B illustrates the integrated circuit **300** of FIG. **10**A at the block-level.

FIG. **10**C illustrates the integrated circuit **300** of FIG. **10**B at the block-level.

FIG. **11**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the present invention (FIG. **9**B), showing the top-level routing for pin assignment.

FIG. **11**B illustrates the integrated circuit **300** of FIG. **11**A at the block-level.

FIG. **11**C illustrates the integrated circuit **300** of FIG. **11**B at the block-level.

FIG. **12**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **12**B illustrates the integrated circuit **300** of FIG. **12**A at the block-level.

FIG. **12**C illustrates the integrated circuit **300** of FIG. **12**B, showing the removal of excess pins.

FIG. **13**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **30** (e.g., routing metal).

FIG. **13**B illustrates the segment **30**A of FIG. **13**A.

FIG. **13**C illustrates the integrated circuit **300** of FIG. **13**A in the top-level, showing that the segment **30**A has been removed from the top-level netlist and merged into the block-level netlist of block**1** **10**.

FIG. **14**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **60** (e.g., routing metal).

FIG. **14**B illustrates the integrated circuit **300** at the block-level.

FIG. **14**C illustrates the integrated circuit **300** at the block-level.

FIG. **15**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **15**B illustrates that the port F of block**1** **10** has been removed from the top-level netlist.

FIG. **16**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **16**B illustrates that the port B of block**1** **10** has been removed from the netlist for the top-level routing for pin assignment.

FIG. **17**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **17**B illustrates that the port F of block**1** **10** has been removed from the top-level netlist.

The drawings referred to in this description should not be understood as being drawn to scale except if specifically noted.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be recognized by one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

Notation and Nomenclature

Some portions of the detailed descriptions which follow are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. In the present application, a procedure, logic block, process, etc., is conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proved convenient at times, principally for reasons of common usage, to refer to these signals a s bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, a variety of terms are discussed that refer to the actions and processes of an electronic system or a computer system, or other electronic computing device/system. The computer system or similar electronic computing device manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices. The present invention is also well suited to the use of other computer systems such as, for example, optical , mechanical, or quantum computers.

Exemplary Computer System Environment

Aspects of the present invention are discussed in terms of steps executed on a computer system. Although a variety of different computer systems can be used with the present invention, an exemplary computer system **200** is shown in FIG. **2**.

With reference to FIG. **2**, portions of the present invention are comprised of computer-readable and computer executable instructions which reside, for example, in computer-usable media of an electronic system such as the exemplary computer system. FIG. **2** illustrates an exemplary computer system **200** on which embodiments of the present invention may be practiced. It is appreciated that the computer system **200** of FIG. **2** is exemplary only and that the present invention can operate within a number of different computer systems including general-purpose computer systems and embedded computer systems.

Computer system **200** includes an address/data bus **110** for communicating information, a central processor **101** coupled with bus **110** for processing information and instructions, a volatile memory **102**. (e.g., random access memory RAM) coupled with the bus **110** for storing information and instructions for the central processor **101** and a non-volatile memory **103** (e.g., read only memory ROM) coupled with the bus **110** for storing static information and instructions for the processor **101**. Exemplary computer system **200** also includes a data storage device **104** ("disk subsystem") such as a magnetic or optical disk and disk drive coupled with the bus **110** for storing information and instructions. Data storage device **104** can include one or more removable magnetic or optical storage media (e.g., diskettes, tapes) which are computer readable memories. Memory units of computer system **200** include volatile memory **102**, non-volatile memory **103** and data storage device **104**.

Exemplary computer system **200** can further include an optional signal generating device **108** (e.g., a network interface card "NIC") coupled to the bus **110** for interfacing with other computer systems. Also included in exemplary computer system **200** of FIG. **2** is an optional alphanumeric input device **106** including alphanumeric and function keys coupled to the bus **110** for communicating information and command selections to the central processor **101**. Exemplary computer system **200** also includes an optional cursor control or directing device **107** coupled to the bus **110** for communicating user input information and command selections to the central processor **101**. An optional display device **105** can also be coupled to the bus **110** for displaying information to the computer user. Display device **105** may be a liquid crystal device, other flat panel display, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters recognizable to the user. Cursor control device **107** allows the user to dynamically signal the two-dimensional movement of a visible symbol (cursor) on a display screen of display device **105**. Many implementations of cursor control device **107** are known in the art including a trackball, mouse, touch pad, joystick or special keys on alphanumeric input device **106** capable of signaling movement of a given direction or manner of displacement. Alternatively, it will be appreciated that a cursor can be directed and/or activated via input from alphanumeric input device **106** using special keys and key sequence commands.

Abutted-pin Hierarchical Physical Design

FIG. **3** illustrates an integrated circuit **300** generated with software tools according to the abutted-pin hierarchical physical design process of the present invention. The abutted-pin hierarchical physical design provides solutions

to the problems of the traditional hierarchical physical design (see FIG. 1) and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced to instantiations of each block **10**–**30** and **60**–**94**.

As illustrated in FIG. **3**, the abutted-pin hierarchical physical design **300** includes a plurality of blocks **10**–**30** and **60**–**94**. The netlist of the integrated circuit **300** is partitioned into the plurality of blocks **10**–**30** and **60**–**94** such that each block **10**–**30** and **60**–**94** has a block level netlist. Blocks **10**–**30** have the major or core components of the integrated circuit **300**. Blocks **60**–**94** have the bonding pads and other support circuitry of the integrated circuit **300**. The blocks **10**–**30** and **60**–**94** can be rectangular in shape and can be rectilinear in shape. It should be understood that the integrated circuit **300** can have any number of blocks.

Each block **10**–**30** and **60**–**94** has one or more pins **50**, whereas each pin **50** represents a location where a signal can enter the block **10**–**30** and **60**–**94** or a location where a signal can exit the block **10**–**30** and **60**–**94**. The edge or boundary of each block **10**–**30** and **60**–**94** rests against the edge or boundary of another block **10**–**30** and **60**–**94**, such that the pin **50** of one block abuts the pin **50** of another block.

Moreover, the top-level components or objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) are not visible because they have been merged into the blocks **10**–**30** and **60**–**94** by a "press" operation performed by a software tool. First, the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) are placed and routed at the top-level (the top-level is shown in FIG. **3**). In the "press" operation, the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) that are within the boundary of a block **10**–**30** and **60**–**94** are removed from the top-level netlist and merged into the block-level netlist of that block **10**–**30** and **60**–**94**. Hence, the abutted-pin hierarchical physical design **300** can be optimized by separately optimizing the individual blocks **10**–**30** and **60**–**94**. Thus, the software tools can generate (e.g., perform placement, routing, timing, verification, etc.) and optimize the individual blocks **10**–**30** and **60**–**94** in parallel. Moreover, a bug within an individual block **10**–**30** and **60**–**94** can be corrected by returning that individual block to the logic design phase, while the other blocks continue to undergo the physical design phase.

FIG. **4** illustrates the abutted-pin hierarchical physical design process **400** according to an embodiment of the present invention. At **410**, a software tool receives the netlist of the integrated circuit from the logic design phase, as described above. The netlist is partitioned into a plurality of blocks, each block having a block-level netlist. In an embodiment, the partitioning of the netlist focuses on reducing the number of ports or terminals of a block that need to couple to the ports or terminals of other blocks.

At **420**, a software tool performs top-level floor planning. Here, the layout of each block is determined. At the end of the top-level floor planning, the top-level for an integrated circuit **300** (as shown in FIG. **6**) is generated. As illustrated in FIG. **6**, the layout of the blocks **10**–**30** is established. In FIG. **6**, the bonding pads **60**–**94** (of FIG. **3**) have been omitted.

At **430**, software tools perform top-level placement and routing for the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.). FIG. **7** illustrates a clock wire **320** and a power wire **310** of the top-level. The clock wire **320** is routed over BlockA **10** and BlockC **30**. The power wire **310** is routed over BlockA **10**. It should be understood that any number of additional top-level objects can be placed and routed at the top-level.

At **440**, a software tool performs a top-level route for obtaining the pin assignments for each block **10**–**30**, as illustrated in FIG. **8**. Since each block **10**–**30** has one or more ports or terminals **47** that needs to couple to a port or terminal of another block **10**–**30**, the pins for each block **10**–**30** have to be defined. Initially, the ports **47** of each block **10**–**30** are placed in a general random location within each block at the top-level since the actual location of the port **47** is not known until a placement operation is performed at the block-level. As illustrated in FIG. **8**, the location **45A**–**45F** where a routing wire **48** crosses a boundary between two blocks is defined as a pin for each of the blocks **10**–**30**, facilitating creation of pins that are abutted. In an embodiment, a software tool creates each pin to have a width that is equivalent to the width of the routing wire **48** at the boundary between the two blocks. The pins **50** are illustrated in FIG. **3**.

At **450**A–**450**C, the abutted-pin hierarchical physical design process **400** enables software tools to generate and to optimize each block **10**–**30** in parallel at the block-level.

FIG. **5** illustrates the abutted-pin hierarchical physical design process **500** as performed at the block-level in a particular block (**450**A–**450**C of FIG. **4**) after step **440** of FIG. **4**.

At **510**, a software tool performs press operations. The top-level objects illustrated in FIG. **7** (e.g., a clock wire **320** and a power wire **310**) and which are located within the boundary of a particular block, are pressed into the particular block. In particular, the top-level objects that are within the boundary of a particular block are removed from the top-level netlist and merged into the block-level netlist of that particular block. Moreover, the pins for the particular block are generated based on the location where the routing wire crosses the boundary between two blocks, as illustrated in FIG. **8** and FIG. **3**.

At **520**, a software tool performs block-level floor planning for the particular block. At **530**, a software tool performs a block-level placement operation for the particular block. At **540**, software tools perform a variety of block-level operations to optimize the particular block. Additionally, at **550**, a block-level route is performed for the particular block by a software tool. At **552** and **554**, software tools perform a block-level extraction operation for determining capacitance and resistance at the nodes and perform block-level timing analysis operations for the particular block.

At **560** and **570**, a variety of software tools perform a number of verification operations such as formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification.

FIG. **9**A illustrates the integrated circuit design flow of the prior art. As illustrated in FIG. **9**A, the physical design phase **910** receives the netlist from the logic design phase (not shown). The physical design phase **910** generates the physical design for the integrated circuit and outputs a GDS II file. The GDS II file is received by the semiconductor factory **920**. The integrated circuit is fabricated by the semiconductor factory **920** on a semiconductor chip.

FIG. **9**B illustrates the integrated circuit design flow according to an embodiment of the present invention. As

illustrated in FIG. 9B, the physical design phase **910** receives the netlist from the logic design phase (not shown). In addition, the physical design phase **910** receives physical design information **930**, whereas the physical design information **930** can be any information about a prior integrated circuit that has undergone the physical design phase **910**. In an embodiment, the physical design information **930** is stored in a database. For example, the physical design information **930** can be pin assignments of the prior integrated circuit, optimal clock distribution tree of the prior integrated circuit, parasitic extraction data of the prior integrated circuit, locations of obstructions such as a RAM of the prior integrated circuit, identification of congested blocks of the prior integrated circuit, metal resources for the blocks of the prior integrated circuit, or any other information which can facilitate optimizing the current integrated circuit. Thus, the software tools of the physical design phase **910** can customize the current integrated circuit to avoid the problems of the prior integrated circuit and to realize the benefits of the prior integrated circuit.

In the physical design phase **910**, decisions made at the top-level with respect to the top-level objects, significantly influence the creation of problems at the block-level and the optimization operations at the block-level. By using physical design information **930** (concerning the block-level of the prior integrated circuit) at the top-level of the current integrated circuit, the decisions made at the top-level with respect to the top-level objects of the current integrated circuit will be able to reduce the problems present in the prior integrated circuit and will be able to generate solutions to overcome the problems present in the prior integrated circuit, improving the optimization of the abutted-pin hierarchical physical design process of the present invention. Thus, if the physical design information **930** has information about several prior integrated circuits, the current integrated circuit is more likely to be optimized.

In addition, the physical design phase **910** generates the physical design for the integrated circuit and outputs a GDS II file. Moreover, the physical design phase **910** stores physical design information **930** of the current integrated circuit to be used in the physical design phase **910** of a future integrated circuit. The GDS II file is received by the semiconductor factory **920**. The integrated circuit is fabricated by the semiconductor factory **920** on a semiconductor chip.

FIG. **10**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the prior art (FIG. **9**A), showing the top-level routing for pin assignment. The port C of block**1** **10** is routed to port B of block**2** **20**. The port A of block**1** **10** is routed to port D of block**2** **20**. This top-level routing has been performed after ports A–D where placed in a generally random location within each block **10**–**20** at the top-level since the actual locations of the ports A–D are not known until a placement operation is performed at the block-level. Here, the software tools at the top-level do not have access to the physical design information of a prior integrated circuit. The locations **15** and **16** are where the routing metal **18** crosses the boundary between two blocks **10** and **20**.

FIG. **10**B illustrates the integrated circuit **300** of FIG. **10**A. At the block-level, the pins **15**A and **16**A were formed for block**1** **10**. At the block-level, the pins **15**B and **16**B were formed for block**2** **20**, whereas pin **15**A abuts pin **15**B and pin **16**A abuts pin **16**B. The pins **15**A and **15**B were formed at location **15** of FIG. **10**A. The pins **16**A and **16**B were formed at location **16** of FIG. **10**A.

FIG. **10**C illustrates the integrated circuit **300** of FIG. **10**B at the block-level. As illustrated in FIG. **10**C, the block-level placement operation for block**1** **10** placed the ports A and C at locations that are different from the locations used to generate the pin assignments in FIG. **10**A. In addition, the block-level placement operation for block**2** **20** placed the ports B and D at locations that are different from the locations used to generate the pin assignments in FIG. **10**A. Hence, the block-level routing operations for blocks **10** and **20** generated an inefficient amount of routing wire **19** to couple the ports to the pins in each block. In sum, the pin assignment affects the optimization of the routing wire **19**.

FIG. **11**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the present invention (FIG. **9**B), showing the top-level routing for pin assignment. The port C of block**1** **10** is routed to port B of block**2** **20**. The port A of block**1** **10** is routed to port D of block**2** **20**. This top-level routing has been performed after each port A–D where placed in a particular location within each block **10**–**20** at the top-level, whereas the particular location was based on using the physical design information associated with the prior integrated circuit (FIGS. **10**A–**10**C). Here, the software tools at the top-level have access to the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C). The locations **15** and **16** are where the routing metal **18** crosses the boundary between two blocks **10** and **20**.

FIG. **11**B illustrates the integrated circuit **300** of FIG. **11**A at the block-level. At the block-level, the pins **15**A and **16**A were formed for block**1** **10**. At the block-level, the pins **15**B and **16**B were formed for block**2** **20**, whereas pin **15**A abuts pin **15**B and pin **16**A abuts pin **16**B. The pins **15**A and **15**B were formed at location **15** of FIG. **1**A. The pins **16**A and **16**B were formed at location **16** of FIG. **11**A. Here, the pins **15**A and **15**B are associated with ports A and D, unlike FIG. **10**B where pins **15**A and **15**B were associated with ports C and B. Moreover, the pins **16**A and **16**B of FIG. **11**B are associated with ports C and B, unlike FIG. **10**B where pins **16**A and **16**B were associated with ports A and D.

FIG. **11**C illustrates the integrated circuit **300** of FIG. **11**B at the block-level. As illustrated in FIG. **11**C, the block-level placement operation for block**1** **10** placed the ports A and C at locations that are different from the locations used to generate the pin assignments in FIG. **11**A. In addition, the block-level placement operation for block**2** **20** placed the ports B and D at locations that are different from the locations used to generate the pin assignments in FIG. **11**A. However, the difference in the location of the ports between FIG. **11**A and FIG. **11**C is less than the difference in the location of the ports between FIG. **10**A and FIG. **10**C. Hence, the block-level routing operations for blocks **10** and **20** generated a more efficient amount of routing wire **19** to couple the ports to the pins in each block, compared to FIG. **10**C. In sum, the pin assignments generated with the use of the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C) were more optimal than the pin assignments generated without the use of the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C).

FIG. **12**A illustrates an integrated circuit **300** based on the abutted-pin a hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. In the course of routing source port **24** of block**3** **30** to destination port **22** of block**2** **20**, the software tool that performs the top-level routing for pin assignment crosses the boundary between block**1** **10** and block**2** **20** at locations **15**,**16**, and **17**, whereas the locations **15**,**16**, and **17** will be defined as pins. The software tool is concerned with routing a path between the source port **24** and the destination port

22, but is not concerned about the number of times the path crosses the boundary between the same blocks.

FIG. 12B illustrates the integrated circuit **300** of FIG. 12A at the block-level. The pins **15A–15B**, **16A–16B**, and **17A–17B** are formed between block1 **10** and block2 **20**. The pins **18A–18B** are formed between block1 **10** and block3 **30**. The presence of pins **16A–16B** and **17A–17B** causes additional routing metal to be added to block1 **10** and block2 **20** so that pins **15A**, **16A**, and **17A** can be coupled within block1 **10** and so that pins **15B**, **16B**, and **17B** can be coupled within block2 **20**. Hence, one pair of pins (**15A–15B** or **16A–16B** or **17A–17B**) is sufficient.

FIG. 12C illustrates the integrated circuit **300** of FIG. 12B, showing the removal of excess pins. As illustrated in FIG. **120**, excess pins **16A–16B** and **17A–17B** were removed from block1 **10** and block2 **20**. This removal is based on a plurality of criteria, such as the current flow direction between the source port **24** and the destination port **22**, the location of the excess pins relative to the source port **24** and the destination port **22**, or any other criteria. Here, the criteria kept pins **15A–15B** but deleted pins **16A–16B** and **17A–17B**.

FIG. **13A** illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **30** (e.g., routing metal). As described above, a software tool performs a press operation so that the portion of the top-level object **30** which is within the boundary of a particular block **10–20** is moved from the top-level netlist to the block-level netlist of the particular block **10–20**. In particular, the segment **30A** is pressed into block1 **10** while the segment **30B** is pressed into block2 **20**. In an embodiment, the shape operations of a database are utilized in performing the press operation. In FIG. **13A**, an AND operation would be performed with block1 **10** and the shape **30** to obtain the segment **30A** (FIG. **13B**). Typically, the routing metal **30** includes a plurality of properties that are stored in a database. These properties identify the routing metal **30** and describe the function of the routing metal **30**. However, in the shape operations (e.g., AND) of the prior art, the shape operation returns the segment **30A** (FIG. **13B**) without its properties. Thus, these properties have to be reconstructed.

In the present invention, the software tool t hat performs the press operation preserves the properties associated with segment **30A** of the routing metal **30** despite the shape operation (e.g., AND) performed with block1 **10** and the shape **30** to obtain the segment **30A** (FIG. 13B).

FIG. 13C illustrates the integrated circuit **300** of FIG. 13A in the top-level, showing that the segment **30A** has been removed from the top-level netlist and merged into the block-level netlist of block1 **10**. Moreover, the properties associated with segment **30A** at the top level are transferred to the segment **30A** at the block-level.

FIG. **14A** illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **60** (e.g., routing metal). As illustrated in FIG. **14A**, the top-level object **60** is routed through block1 **10**, block2 **20**, and block3 **30**. The locations **51–52** indicate top-level object **60** crosses a boundary between two blocks. In an embodiment, a press property is added to the properties of the top-level object **60** stored in a database. If the top-level object **60** has the press property, the top-level object **60** retains its location when the top-level object **60** is pressed into block1 **10**, block **20**, and block3 **30**, as illustrated in the block-level view of the integrated circuit **300** in FIG. **14C**. If the top-level object **60** does not have the press property, the top-level object **60** generally does not retain its location when the top-level object **60** is pressed into block1 **10**, block2 **20**, and block3 **30**, as illustrated in the block-level view of the integrated circuit **300** in FIG. **14B**. For example, top-level objects such as power and ground have the press property. As illustrated in FIG. **14B**, the pins **51A–51B** and **52A–52B** are defined. However, the software tool is not constrained to placing the top-level object **60** in the block-level exactly as it was placed at the top-level. Moreover, the top-level object is placed in the block-level of block1 **10**, block2 **20**, and block3 **30** according to the separate placement and routing requirements of block1 **10**, block2 **20**, and block3 **30**. –FIG. **15A** illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **15A**, in the top-level netlist, the instantiation of block1 **10** includes port F that is unused, thus, not needed for the top-level outing for pin assignment. Hence, a software tool removes port F from the top-level netlist, but the block-level netlist of block1 **10** remains unchanged. In an embodiment, the software tool that performs the press operation removes the port F. FIG. **15B** illustrates that the port F of block1 **10** has been removed from the top-level netlist.

FIG. **16A** illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **16A**, port F and port B of block1 **10** are coupled to port C of block2 **20** with a routing metal **40**. However, at location **30** the routing metal **40** crosses the boundary between block1 **10** and block2 **20**. If a pin is formed within block1 **10** at location **30**, the pin would be coupled to port F and to port B. However, some software tools are not able to represent this relationship (i.e., more than one port coupled to a pin). Hence, a software tool removes one of the ports (port F or port B) from the netlist based on some criteria, such as whether a port is an input port or an output port. FIG. **16B** illustrates that the port B of block1 **10** has been removed from the netlist for the top-level routing for pin assignment.

FIG. **17A** illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **17A**, in the top-level netlist, the instantiation of block1 **10** includes a port F that is tied to either the power line (**1**) or the ground line (**0**) rather to a port of another block. Hence, a software tool removes port F from the top-level netlist to avoid routing the port F at the top-level. Moreover, the software tool ties the port F to either power line (**1**) or the ground line (**0**) in the block-level netlist of block1 **10**. FIG. **17B** illustrates that the port F of block1 **10** has been removed from the top-level netlist.

As illustrated in FIG. **3**, the integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention includes a North bond pad block **60**, an East bond pad block **70**, a South bond pad block **80**, and a West bond pad block **90**, each having bond pad cells. The top-level netlist of the integrated circuit **300** includes one or more top-level inputs for receiving external signals and one or more top-level outputs for transmitting signals off the chip. The top-level inputs and the top-level outputs are coupled to bond pad cells. Typically, software tools which perform a routing operation are configured to not perform the routing operation if the netlist includes bond pad cells.

Since the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** have bond pad cells in the block-level netlist, the software tools refuse to perform the routing operation in these blocks, preventing pins to be formed on the boundary between these blocks and the blocks **10–30** (the core blocks).

In the present invention, the bond pad cells are marked as macrocells rather than bond pad cells, allowing pins to be formed on the boundary between these blocks **60**, **70**, **80**, and **90** and the blocks **10–30** (the core blocks).

Typically, the block-level netlist of the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** include nets to the top-level inputs and nets to the top-level outputs. Generally, the block-level netlist of the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** include nets to the bond pad cells.

In an embodiment of the present invention, a software tool removes the nets to the top-level inputs and nets to the top-level outputs so that the physical design of the integrated circuit can be accomplished as described above. In an embodiment, the software tool removes in the block-level netlist the ports that couple to the top-level inputs and to the top-level outputs. Moreover, the software tool adds a property to the nets to the bond pad cells to indicate that these nets are suppose to couple to the top-level inputs and to the top-level outputs, facilitating an unwinding operation to re-establish at the block-level netlist the nets to the top-level inputs and nets to the top-level outputs that were removed earlier. The un winding operation adds to the block-level netlist the ports (which were removed earlier) that couple to the top-level inputs and to the top-level outputs. Thus, the netlist modified by the physical design phase (e.g., repeater and buffers are added to the netlist) can be compared with the netlist originally received from the logic design phase. In particular, formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification can be performed by software tools.

A challenge with implementing an integrated circuit based on the abutted-pin hierarchical physical design process of the present invention involves analyzing the timing of signal paths that traverse more than one block. The timing of these global paths is difficult to analyzed compared to analyzing the timing of local paths, whereas local paths are signal paths that do not leave a block. One method of analyzing the timing of these global paths involves partitioning the block-level netlist of each block into a first netlist and a second netlist. The first netlist includes nets which start at a register (or flip-flop) and end at a register (or flip-flop) within the block, whereas each branch of the net also starts at a register (or flip-flop) and ends at a register (or flip-flop) within the block. The second netlist includes nets which are coupled to a pin of the block. Generally, the first netlist is ¾ of the initial block-level netlist while the second netlist is ¼ of the initial block-level netlist. If the second netlist ratio is greater than ¼, this indicates inefficient partitioning of the blocks.

Once the first netlist and the second netlist are obtain, an extraction operation to obtain parasitic resistance and capacitance is performed on the second netlist of each block. In an embodiment, the partitioning of the block-level netlist and the extraction operation in each block are performed in parallel. Moreover, an extraction operation is performed on the top-level netlist. In an embodiment, a software tool replaces the abutted pins of the top-level netlist with zero ohm resistors.

Some software tools utilized to perform the timing analysis are unable to operate on netlists having nets that are coupled to multiple pins of a block. In an embodiment of the present invention, these netlist are transformed by using "assign statements" to assign different names to the nets that are coupled to multiple pins of a block. Hence, each different named net can be coupled to a separate pin of the block.

In an embodiment, the second netlist and its associated extraction file of each block and the top-level netlist and its associated extraction file are utilized by software tools to perform the timing analysis. This timing analysis can be performed significantly faster than the case where the block-level netlist is not partitioned into the first netlist and the second netlist. In an embodiment, the timing graph resulting from the timing analysis can be analyzed to extract timing constraints (relating to the delay that can be generated by a block) for each block. Hence, if a block is optimized to meet its extracted timing constrains, the block is more likely to meet its timing parameter when the block interacts with the other blocks in the integrated circuit.

The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the Claims appended hereto and their equivalents.

What is claimed is:

1. A method of improving a physical design of a current integrated circuit, comprising the steps of:

a) receiving a netlist of said current integrated circuit;

b) receiving physical design information from a prior integrated circuit; and

c) generating said physical design based on said netlist and said physical design information.

2. A method as recited in claim **1** wherein said physical design information includes pin assignments of blocks of said prior integrated circuit.

3. A method as recited in claim **1** wherein said physical design information includes optimal clock distribution tree of said prior integrated circuit.

4. A method as recited in claim **1** wherein said physical design information includes parasitic extraction data of said prior integrated circuit.

5. A method as recited in claim **1** wherein said physical design information includes identification of congested blocks of said prior integrated circuit.

6. A method as recited in claim **1** wherein said physical design information includes metal resources of said prior integrated circuit.

7. A method as recited in claim **1** wherein said physical design information includes information which facilitates optimizing said current integrated circuit.

8. A method as recited in claim **1** wherein said step c) includes:

generating a top-level physical design of said current integrated circuit using said netlist and said physical design information including block-level physical design information of said prior integrated circuit.

9. A method as recited in claim **1** wherein said step c) includes:

generating a block-level physical design of said current integrated circuit using said netlist and said physical design information including block-level physical design information of said prior integrated circuit.

10. A method as recited in claim 1 wherein said physical design is an abutted-pin hierarchical physical design including a top-level physical design and a block-level physical design.

11. A method as recited in claim 1 wherein said physical design information includes locations of obstructions of said prior integrated circuit.

12. A method as recited in claim 11 wherein said obstructions include a random access memory (RAM).

13. A method as recited in claim 1 wherein said step c) includes:

partitioning said netlist into a plurality of blocks, each block including a block-level netlist;

performing a top-level floor planning;

performing a top-level placement and route for a plurality of top-level objects;

performing a top-level placement and route for a plurality of ports from said blocks to determine pin assignments for each block; and

generating and optimizing a block-level physical design for each block in parallel.

14. A method as recited in claim 13 wherein said generating and optimizing includes:

pressing each portion of each top-level object, which is located within a boundary of a particular block, into said particular block;

generating each pin for each block based on said top-level placement and route to determine pin assignments;

performing a block-level floor planning for each block;

performing a block-level placement for each block;

performing a plurality of block-level operations to optimize each block; and

performing a block-level route for each block.

15. A computer-readable medium comprising computer-executable instructions stored therein for performing a method of improving a physical design of a current integrated circuit, said method comprising:

a) receiving a netlist of said current integrated circuit;

b) receiving physical design information from a prior integrated circuit; and;

c) generating said physical design based on said netlist and said physical design information.

16. A computer-readable medium as recited in claim 15 wherein said physical design information includes pin assignments of blocks of said prior integrated circuit.

17. A computer-readable medium as recited in claim 15 wherein said physical design information includes optimal clock distribution tree of said prior integrated circuit.

18. A computer-readable medium as recited in claim 15 wherein said physical design information includes parasitic extraction data of said prior integrated circuit.

19. A computer-readable medium as recited in claim 15 wherein said physical design information includes identification of congested blocks of said prior integrated circuit.

20. A computer-readable medium as recited in claim 15 wherein said physical design information includes metal resources of said prior integrated circuit.

21. A computer-readable medium as recited in claim 15 wherein said physical design information includes information which facilitates optimizing said current integrated circuit.

22. A computer-readable medium as recited in claim 15 wherein said step c) includes:

generating a top-level physical design of said current integrated circuit using said netlist and said physical design information including block-level physical design information of said prior integrated circuit.

23. A computer-readable medium as recited in claim 15 wherein said step c) includes:

generating a block-level physical design of said current integrated circuit using said netlist and said physical design information including block-level physical design information of said prior integrated circuit.

24. A computer-readable medium as recited in claim 15 wherein said physical design is an abutted-pin hierarchical physical design including a top-level physical design and a block-level physical design.

25. A computer-readable medium as recited in claim 15 wherein said physical design information includes locations of obstructions of said prior integrated circuit.

26. A computer-readable medium as recited in claim 25 wherein said obstructions include a random access memory (RAM).

27. A computer-readable medium as recited in claim 15 wherein said step c) includes:

partitioning said netlist into a plurality of blocks, each block including a block-level netlist;

performing a top-level floor planning;

performing a top-level placement and route for a plurality of top-level objects;

performing a top-level placement and route for a plurality of ports from said blocks to determine pin assignments for each block; and

generating and optimizing a block-level physical design for each block in parallel.

28. A computer-readable medium as recited in claim 27 wherein said generating and optimizing includes:

pressing each portion of each top-level object, which is located within a boundary of a particular block, into said particular block;

generating each pin for each block based on said top-level placement and route to determine pin assignments;

performing a block-level floor planning for each block;

performing a block-level placement for each block;

performing a plurality of block-level operations to optimize each block; and

performing a block-level route for each block.

29. A method of determining a plurality of pins for each block of a physical design of a current integrated circuit, comprising:

a) receiving a netlist of said current integrated circuit;

b) receiving physical design information from a prior integrated circuit, wherein said physical design information includes pin assignments of blocks of said prior integrated circuit;

c) using said netlist and said physical design information to perform a top-level placement for a plurality of ports corresponding to each block of said current integrated circuit;

d) using said netlist and said physical design information to perform a top-level route for said ports to determine pin assignments for each block of said current integrated circuit; and

e) generating each pin for each block based on said top-level route to determine pin assignments.

30. A method as recited in claim 29 further comprising:

partitioning said netlist into a plurality of blocks of said current integrated circuit, each block including a block-level netlist.

31. A method as recited in claim 29 wherein said physical design information includes optimal clock distribution tree of said prior integrated circuit.

32. A method as recited in claim 29 wherein said physical design information includes parasitic extraction data of said prior integrated circuit.

33. A method as recited in claim 29 wherein said physical design information includes identification of congested blocks of said prior integrated circuit.

34. A method as recited in claim 29 wherein said physical design information includes metal resources of said prior integrated circuit.

35. A method as recited in claim 29 wherein said physical design information includes information which facilitates optimizing said current integrated circuit.

36. A method as recited in claim 29 wherein said physical design information includes locations of obstructions of said prior integrated circuit.

37. A method as recited in claim 36 wherein said obstructions include a random access memory (RAM).

38. A method as recited in claim 29 wherein said physical design is an abutted-pin hierarchical physical design.

39. A method as recited in claim 38 wherein said physical design includes a top-level physical design.

40. A method as recited in claim 38 wherein said physical design includes a block-level physical design.

41. A computer-readable medium comprising computer-executable instructions stored therein for performing a method of determining a plurality of pins for each block of a physical design of a current integrated circuit, comprising:

a) receiving a netlist of said current integrated circuit;

b) receiving physical design information from a prior integrated circuit, wherein said physical design information includes pin assignments of blocks of said prior integrated circuit;

c) using said netlist and said physical design information to perform a top-level placement for a plurality of ports corresponding to each block of said current integrated circuit;

d) using said netlist and said physical design information to perform a top-level route for said ports to determine pin assignments for each block of said current integrated circuit; and

e) generating each pin for each block based on said top-level route to determine pin assignments.

42. A computer-readable medium as recited in claim 41 wherein said method further comprises:

partitioning said netlist into a plurality of blocks of said current integrated circuit, each block including a block-level netlist.

43. A computer-readable medium as recited in claim 41 wherein said physical design information includes optimal clock distribution tree of said prior integrated circuit.

44. A computer-readable medium as recited in claim 41 wherein said physical design information includes parasitic extraction data of said prior integrated circuit.

45. A computer-readable medium as recited in claim 41 wherein said physical design information includes identification of congested blocks of said prior integrated circuit.

46. A computer-readable medium as recited in claim 41 wherein said physical design information includes metal resources of said prior integrated circuit.

47. A computer-readable medium as recited in claim 41 wherein said physical design information includes information which facilitates optimizing said current integrated circuit.

48. A computer-readable medium as recited in claim 41 wherein said physical design information includes locations of obstructions of said prior integrated circuit.

49. A computer-readable medium as recited in claim 48 wherein said obstructions include a random access memory (RAM).

50. A computer-readable medium as recited in claim 41 wherein said physical design is an abutted-pin hierarchical physical design.

51. A computer-readable medium as recited in claim 50 wherein said physical design includes a top-level physical design.

52. A computer-readable medium as recited in claim 50 wherein said physical design includes a block-level physical design.

* * * * *

EXHIBIT 11

US006854093B1

(12) **United States Patent**
Dahl et al.

(10) **Patent No.: US 6,854,093 B1**
(45) **Date of Patent: Feb. 8, 2005**

(54) **FACILITATING PRESS OPERATION IN ABUTTED-PIN HIERARCHICAL PHYSICAL DESIGN**

(75) Inventors: **Peter Dahl**, Cupertino, CA (US); **Byron Dickinson**, San Jose, CA (US); **Margie Levine**, Menlo Park, CA (US); **Paul Rodman**, Palo Alto, CA (US)

(73) Assignee: **Reshape, Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 279 days.

(21) Appl. No.: **10/104,813**

(22) Filed: **Mar. 22, 2002**

Related U.S. Application Data

(63) Continuation of application No. 09/714,722, filed on Nov. 15, 2000.

(51) **Int. Cl.**[7] **G06F 17/50**
(52) **U.S. Cl.** **716/2**; 716/3; 716/7; 716/8
(58) **Field of Search** 716/2, 3, 7, 8

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,475,607 A * 12/1995 Apte et al. 716/10
5,519,628 A * 5/1996 Russell et al. 716/10
5,604,680 A * 2/1997 Bamji et al. 716/8
5,812,415 A * 9/1998 Baisuck 716/11
5,831,869 A * 11/1998 Ellis et al. 716/6
5,943,235 A * 8/1999 Earl et al. 700/98
5,970,238 A * 10/1999 Shibata et al. 716/8
6,009,250 A * 12/1999 Ho et al. 716/5
6,047,116 A * 4/2000 Murakami et al. 716/19
6,145,117 A * 11/2000 Eng 716/18
6,167,555 A * 12/2000 Lakos 716/3
6,167,556 A * 12/2000 Sun et al. 716/3
6,185,727 B1 * 2/2001 Liebmann 716/19
6,243,854 B1 * 6/2001 Lavin et al. 716/19
6,256,768 B1 * 7/2001 Igusa 716/11
6,275,971 B1 * 8/2001 Levy et al. 716/5
6,289,493 B1 * 9/2001 Kita 716/11
6,295,633 B1 * 9/2001 Murakawa 716/8
6,301,698 B1 * 10/2001 Lin et al. 716/19
6,327,695 B1 * 12/2001 Bothra et al. 716/8
6,343,370 B1 * 1/2002 Taoka et al. 716/21
6,453,452 B1 * 9/2002 Chang et al. 716/8
6,507,944 B1 * 1/2003 Kikuchi et al. 716/21
6,567,967 B2 * 5/2003 Greidinger et al. 716/10
6,647,543 B2 * 11/2003 Yamada et al. 716/21
6,665,857 B2 * 12/2003 Ayres 716/19
2002/0000995 A1 * 1/2002 Sawada et al. 345/620
2002/0188914 A1 * 12/2002 Parashkevov et al. 716/3
2003/0018948 A1 * 1/2003 Chang et al. 716/8
2003/0046655 A1 * 3/2003 Kimura 716/19

FOREIGN PATENT DOCUMENTS

JP 07140966 A * 6/1995 ..... G09G/5/36

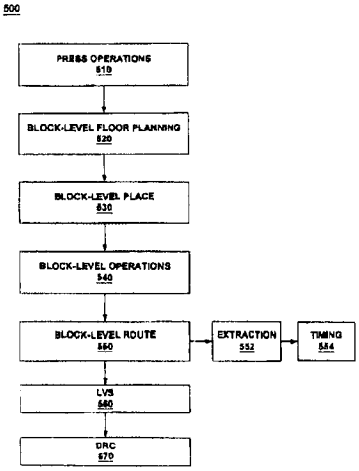
* cited by examiner

Primary Examiner—Matthew Smith
Assistant Examiner—Phallaka Kik
(74) *Attorney, Agent, or Firm*—Wagner, Murabito & Hao LLP

(57) **ABSTRACT**

An abutted-pin hierarchical physical design process is described. The abutted-pin hierarchical physical design provides solutions to the problems of the traditional hierarchical physical design and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced significantly.

40 Claims, 31 Drawing Sheets



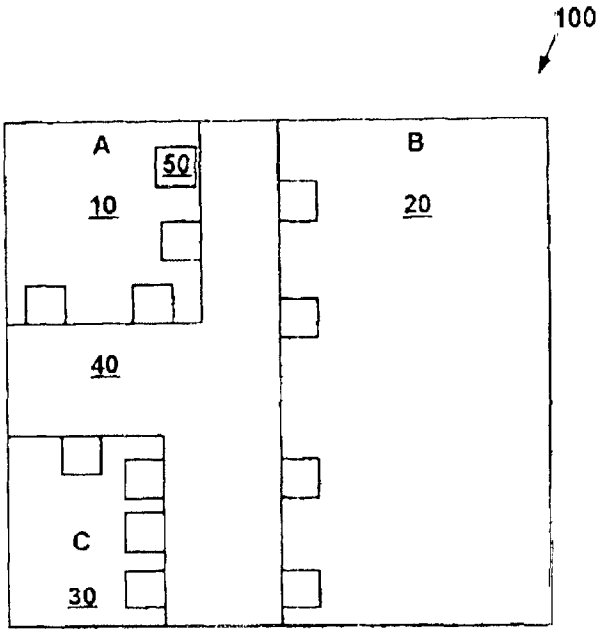


FIGURE 1
(PRIOR ART)

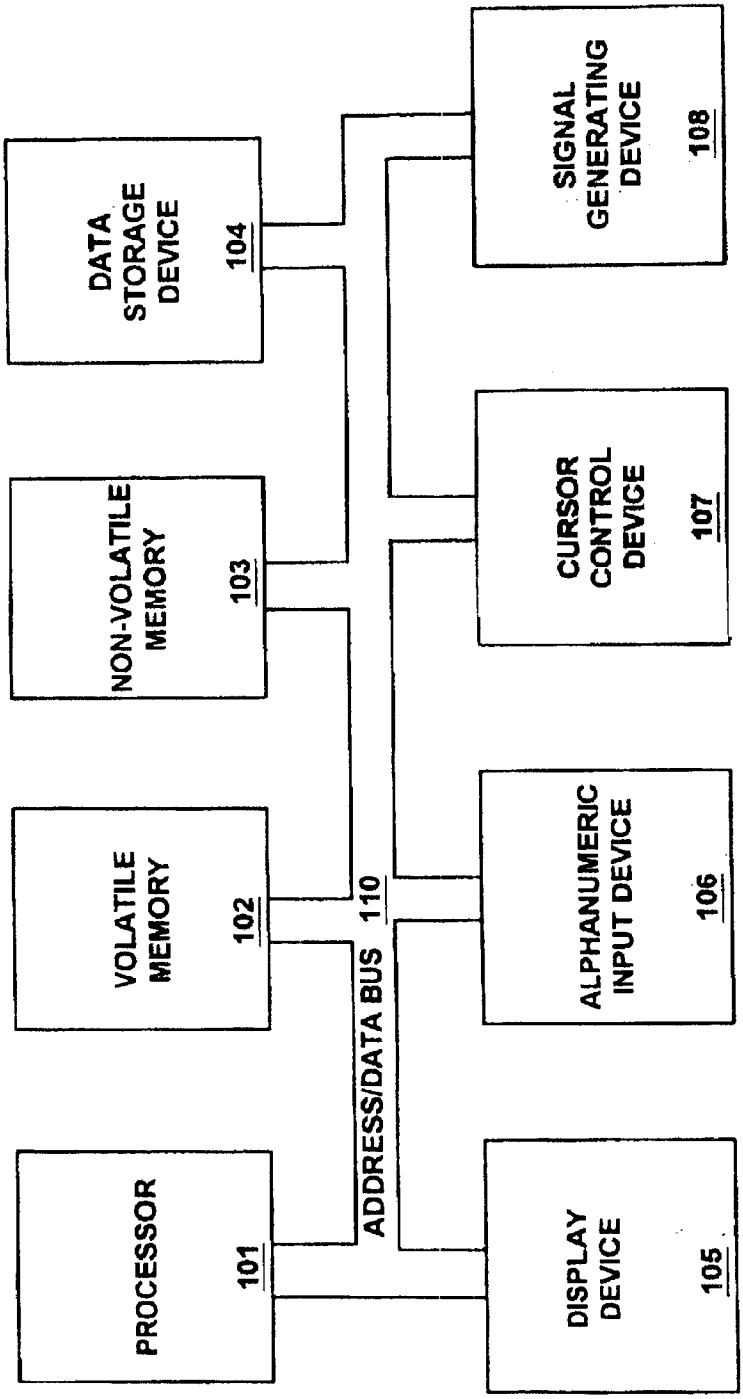

PROCESSOR 101
VOLATILE MEMORY 102
NON-VOLATILE MEMORY 103
DATA STORAGE DEVICE 104
DISPLAY DEVICE 105
ALPHANUMERIC INPUT DEVICE 106
CURSOR CONTROL DEVICE 107
SIGNAL GENERATING DEVICE 108
ADDRESS/DATA BUS 110

FIGURE 2

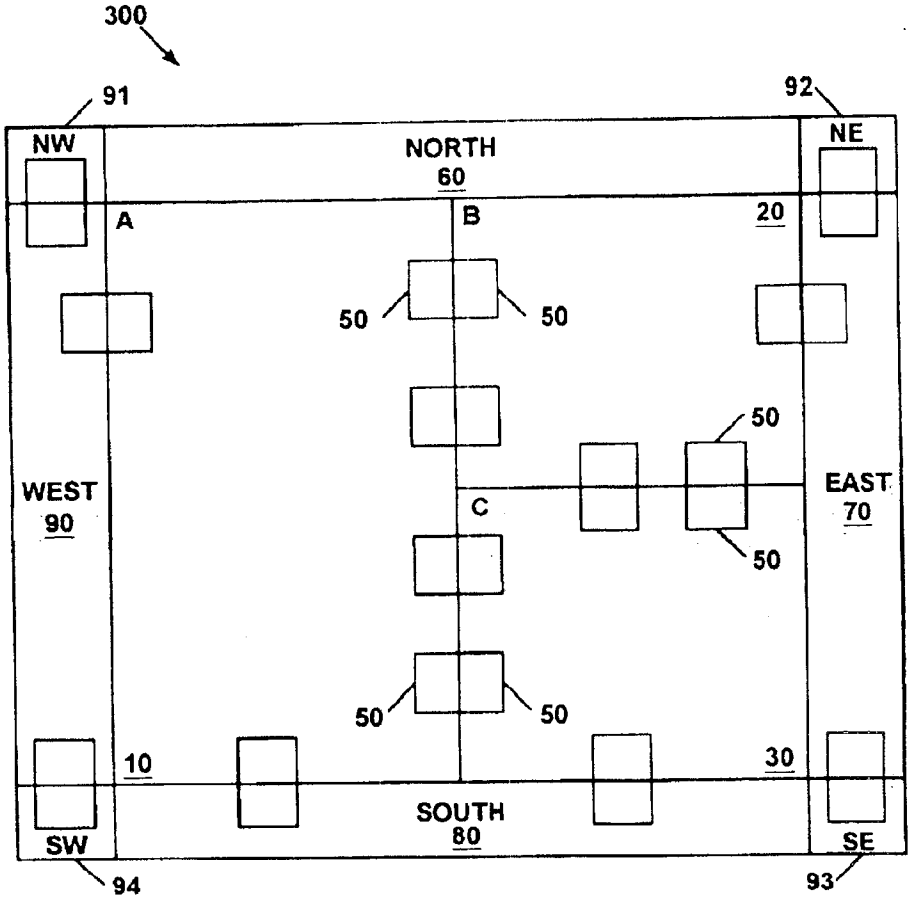

300
91
92
NW
NORTH
60
NE
A
B
20
50
50
50
WEST
90
C
EAST
70
50
50
50
10
30
SOUTH
80
SW
SE
94
93

FIGURE 3

400

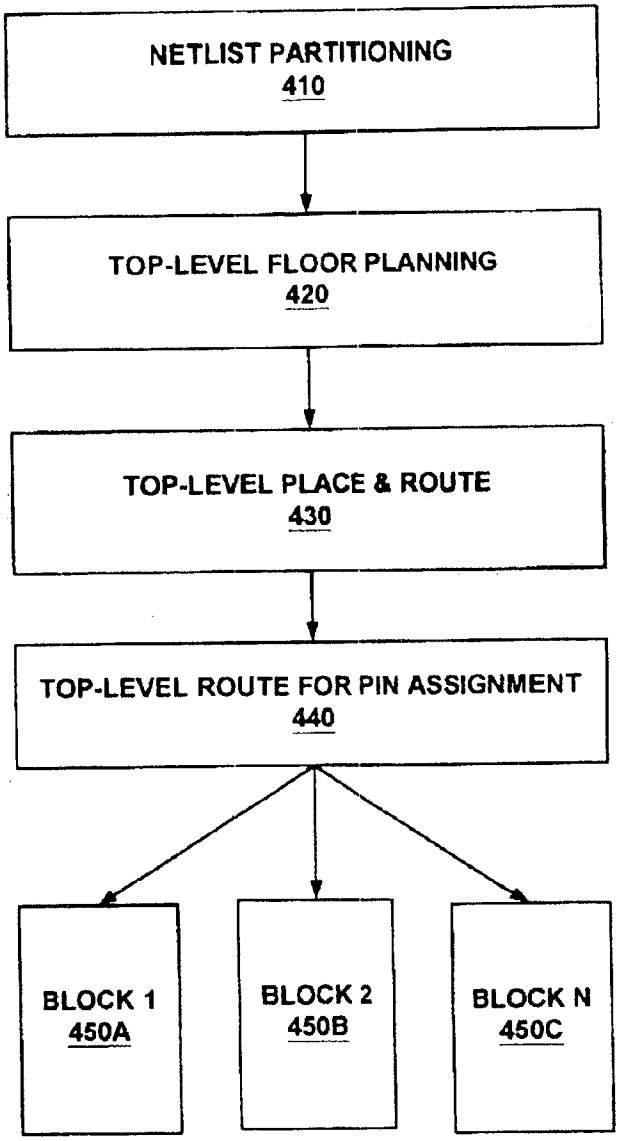

NETLIST PARTITIONING
410
TOP-LEVEL FLOOR PLANNING
420
TOP-LEVEL PLACE & ROUTE
430
TOP-LEVEL ROUTE FOR PIN ASSIGNMENT
440
BLOCK 1
450A
BLOCK 2
450B
BLOCK N
450C

FIGURE 4

500

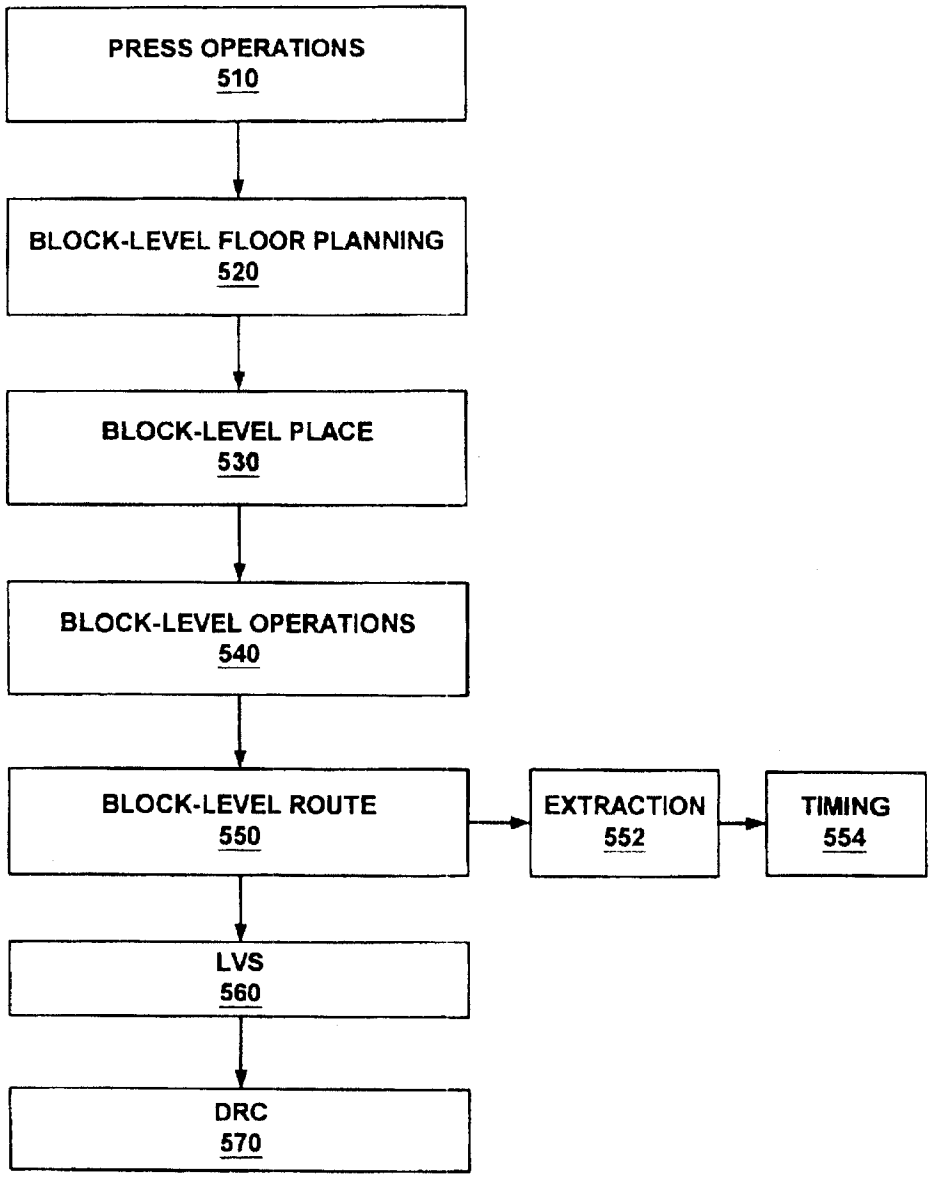

PRESS OPERATIONS
510
BLOCK-LEVEL FLOOR PLANNING
520
BLOCK-LEVEL PLACE
530
BLOCK-LEVEL OPERATIONS
540
BLOCK-LEVEL ROUTE
550
EXTRACTION
552
TIMING
554
LVS
560
DRC
570

FIGURE 5

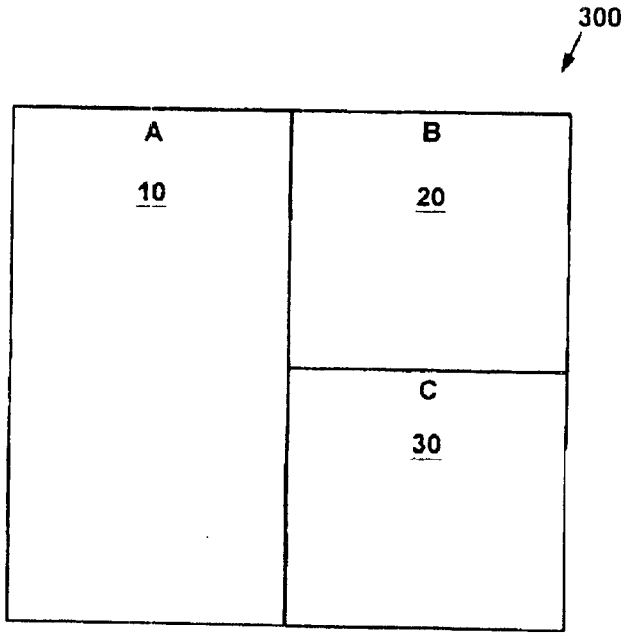


FIGURE 6

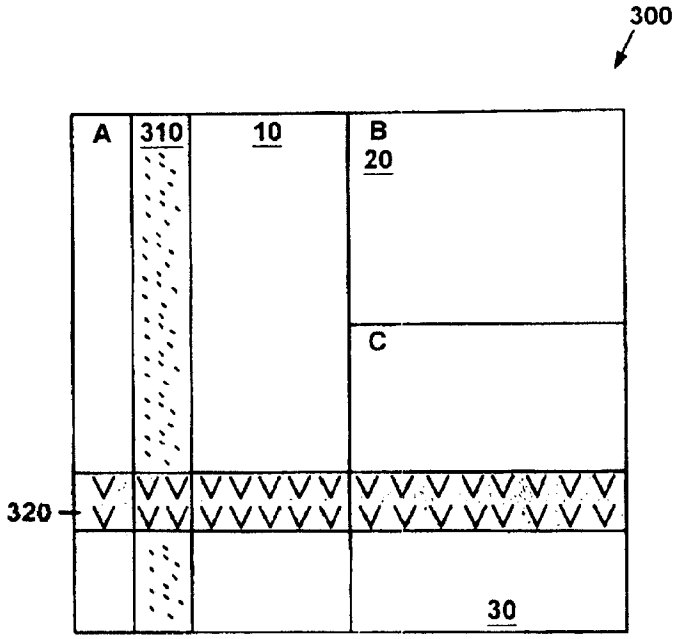


FIGURE 7

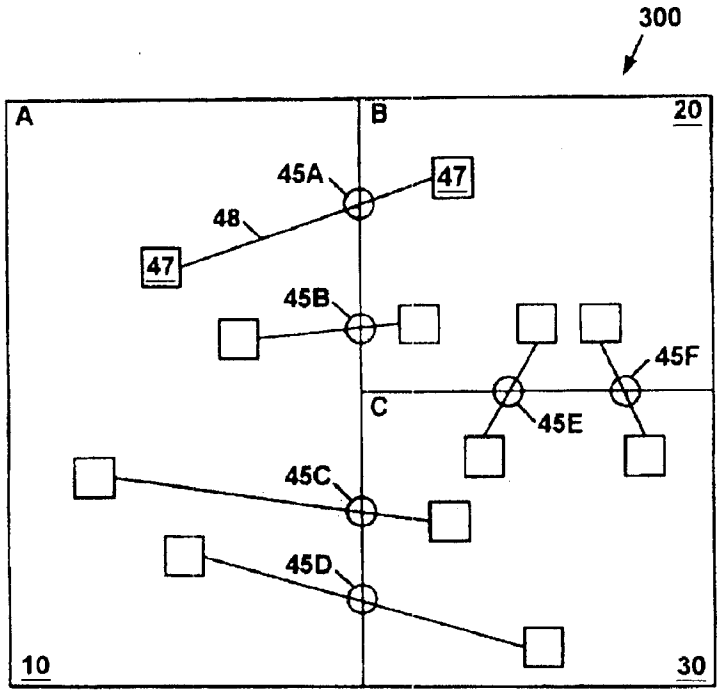


FIGURE 8

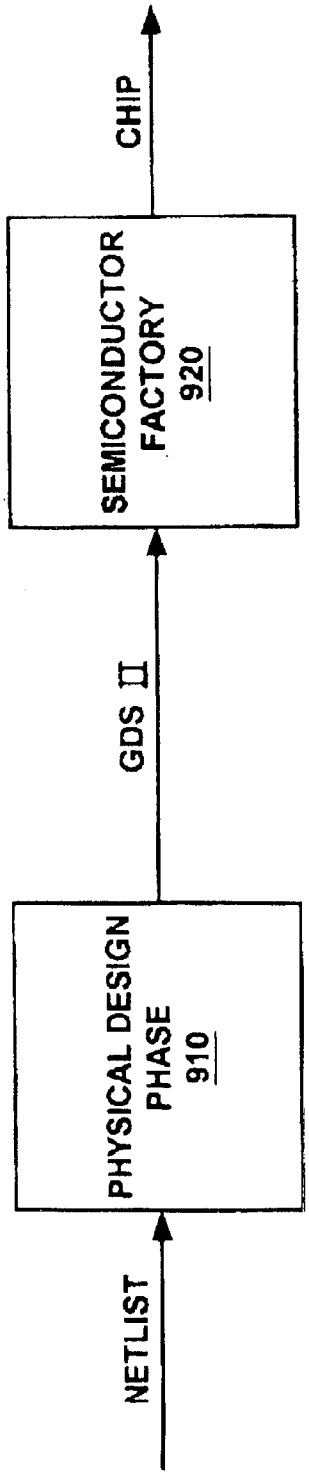


FIGURE 9A
(PRIOR ART)

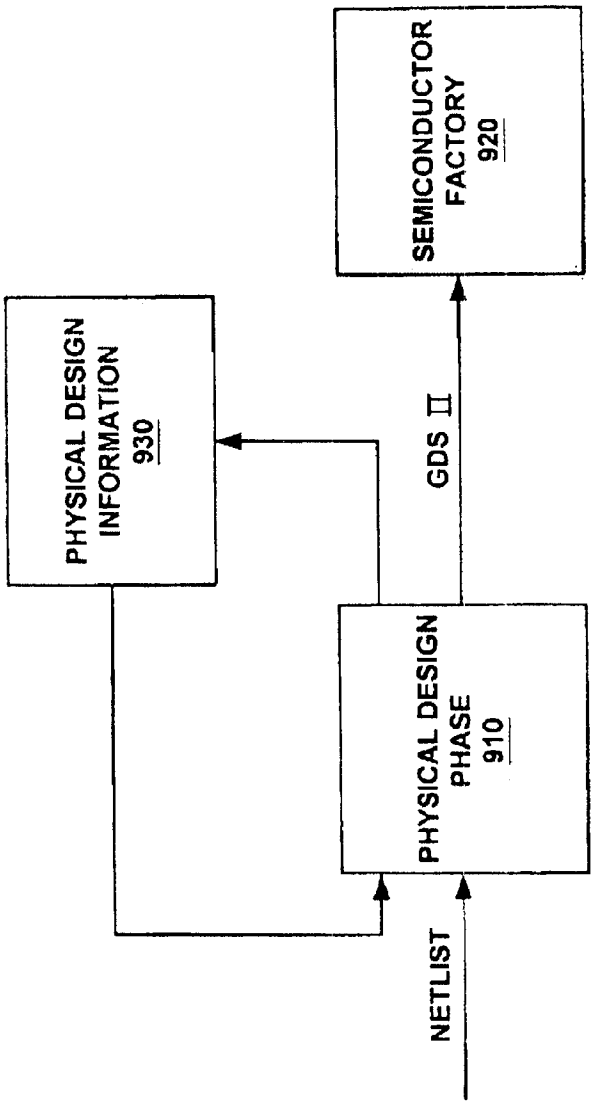

SEMICONDUCTOR FACTORY 920
PHYSICAL DESIGN INFORMATION 930
GDS II
PHYSICAL DESIGN PHASE 910
NETLIST

FIGURE 9B

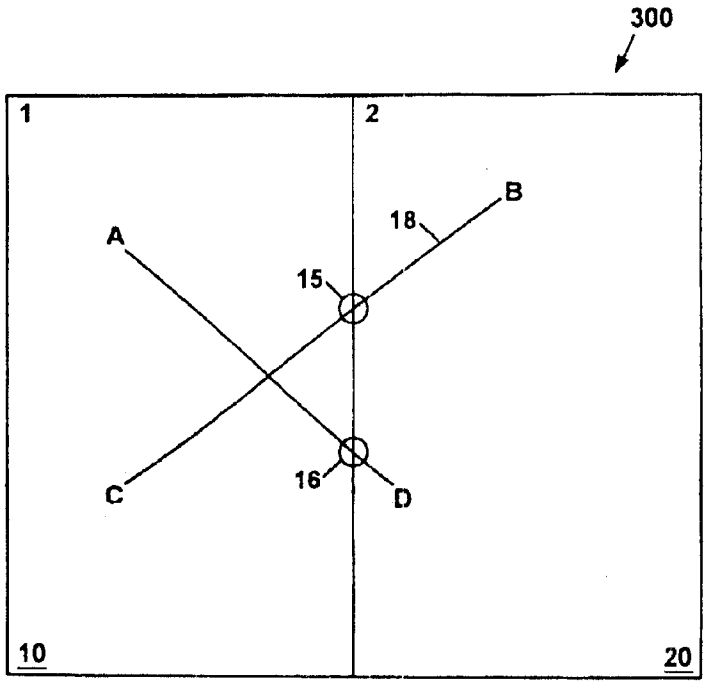


FIGURE 10A

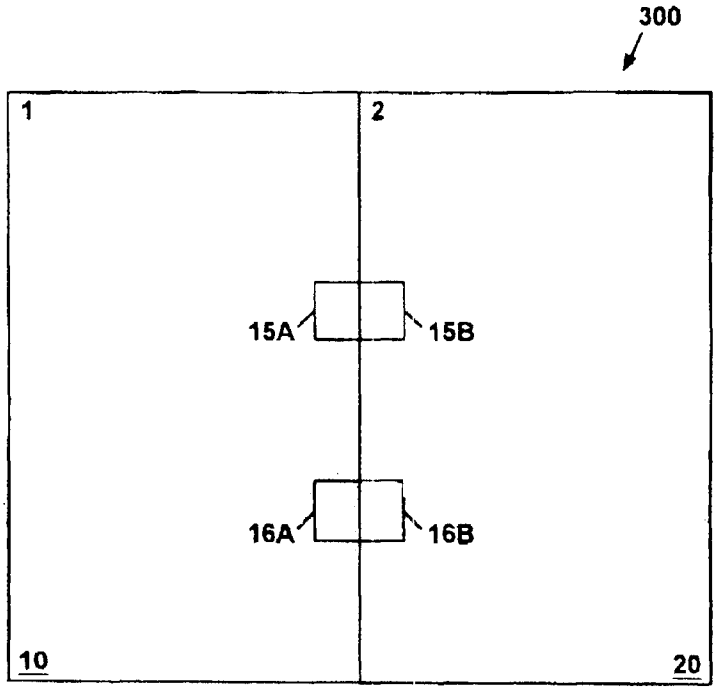


FIGURE 10B

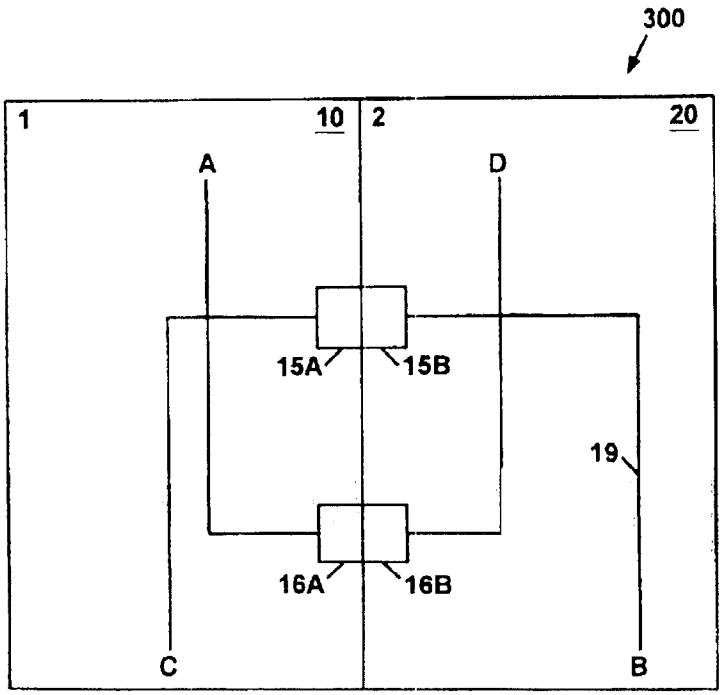


FIGURE 10C

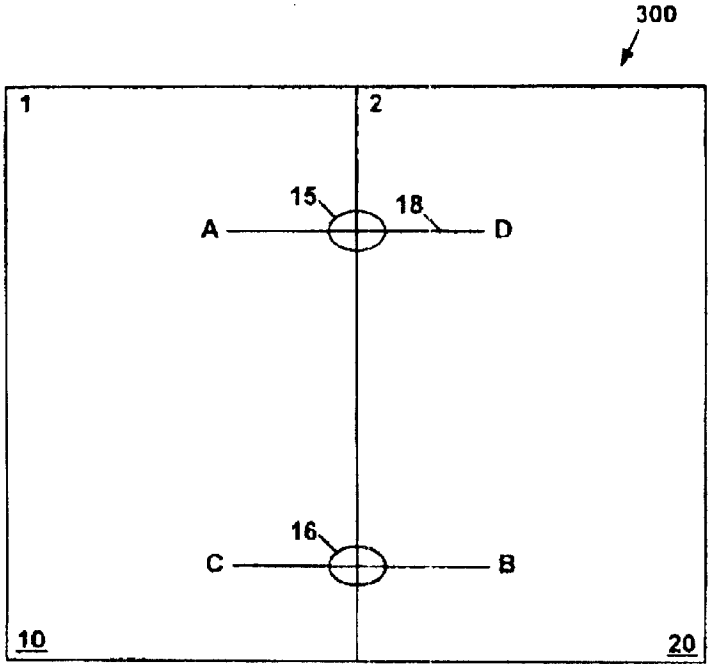


FIGURE 11A

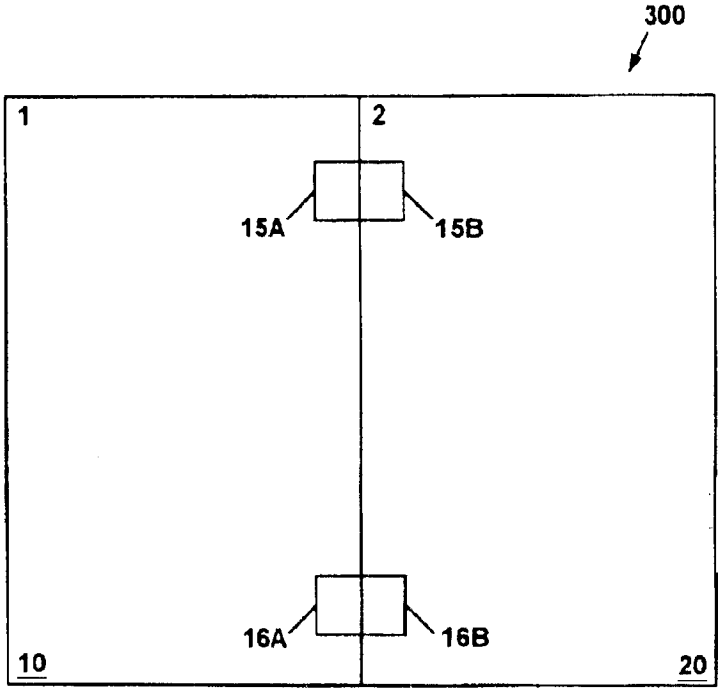


FIGURE 11B

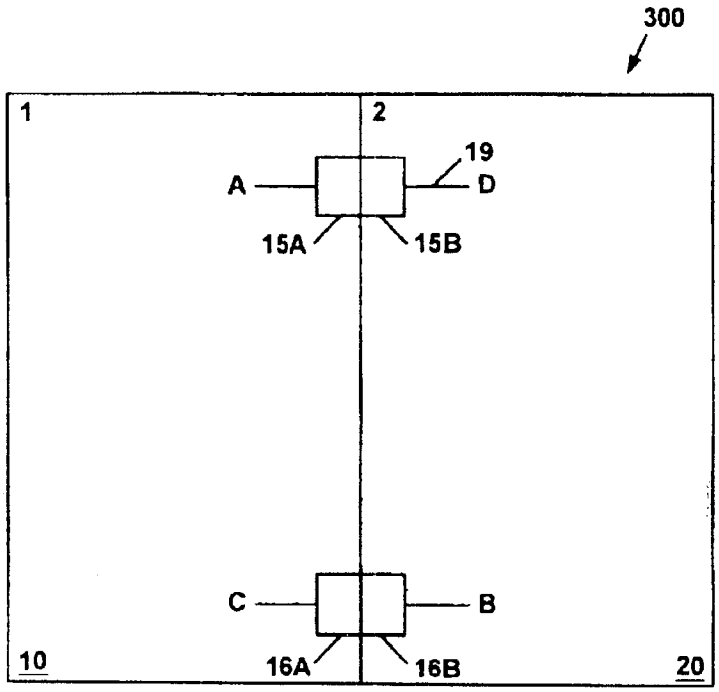


FIGURE 11C

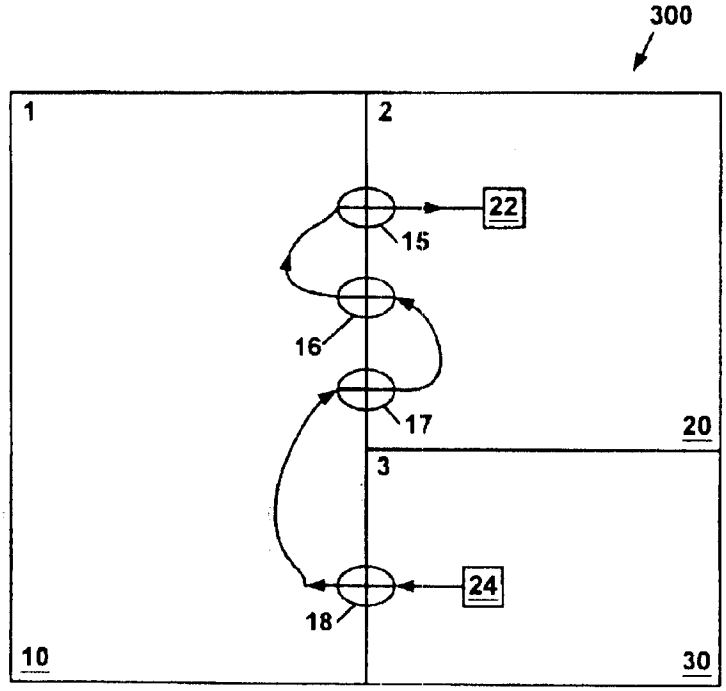


FIGURE 12A

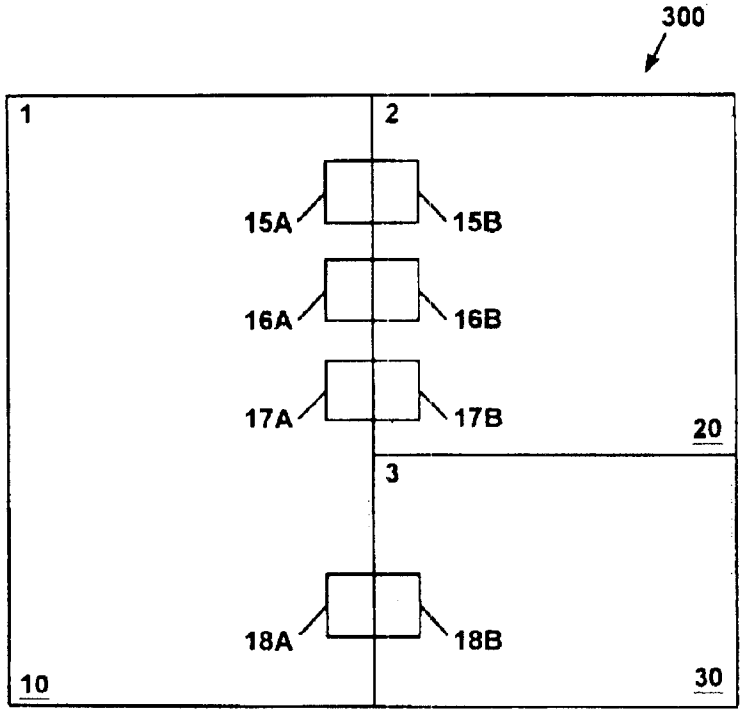


FIGURE 12B

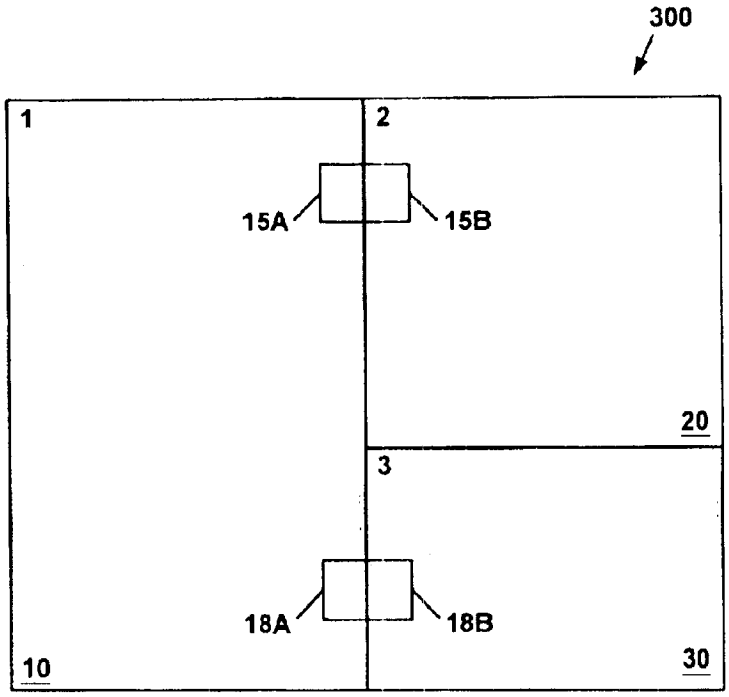


FIGURE 12C

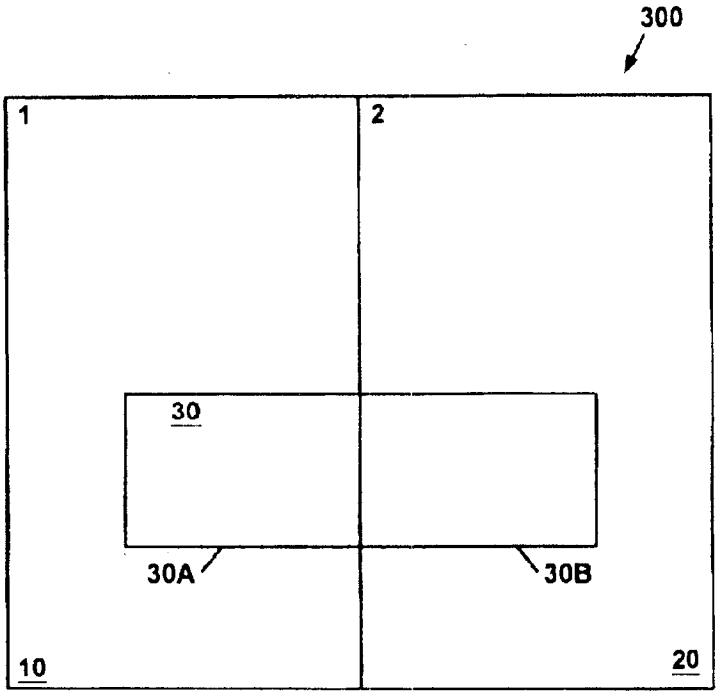


FIGURE 13A

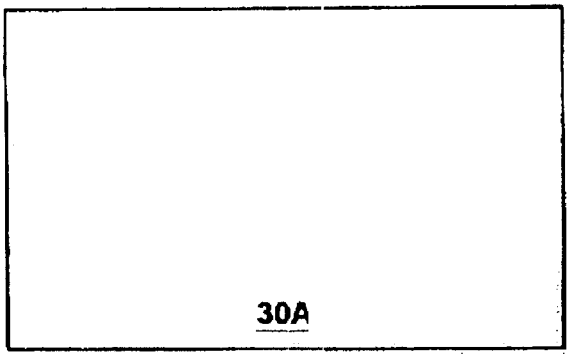


FIGURE 13B

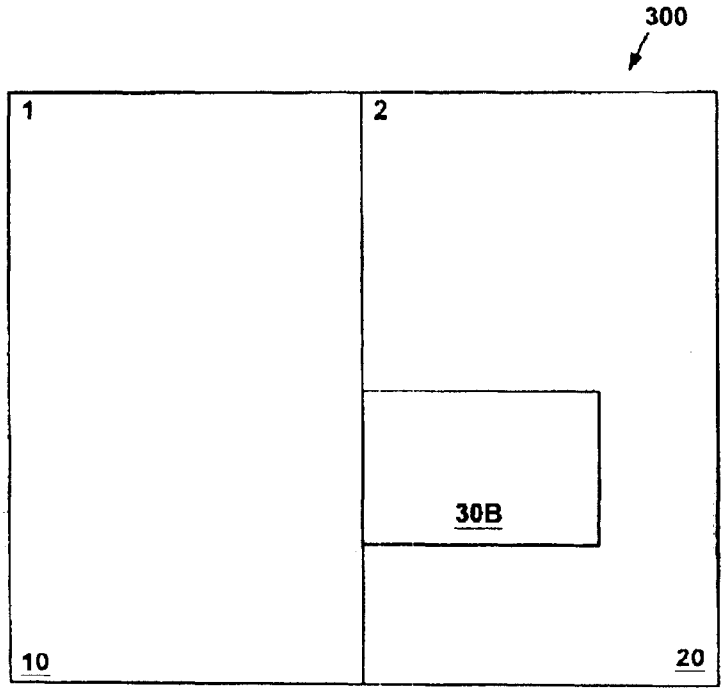


FIGURE 13C

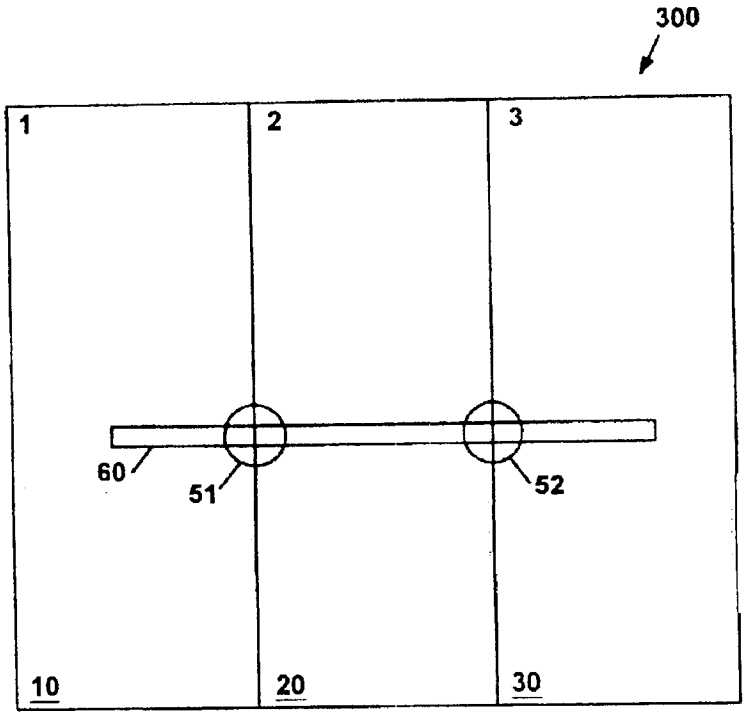


FIGURE 14A

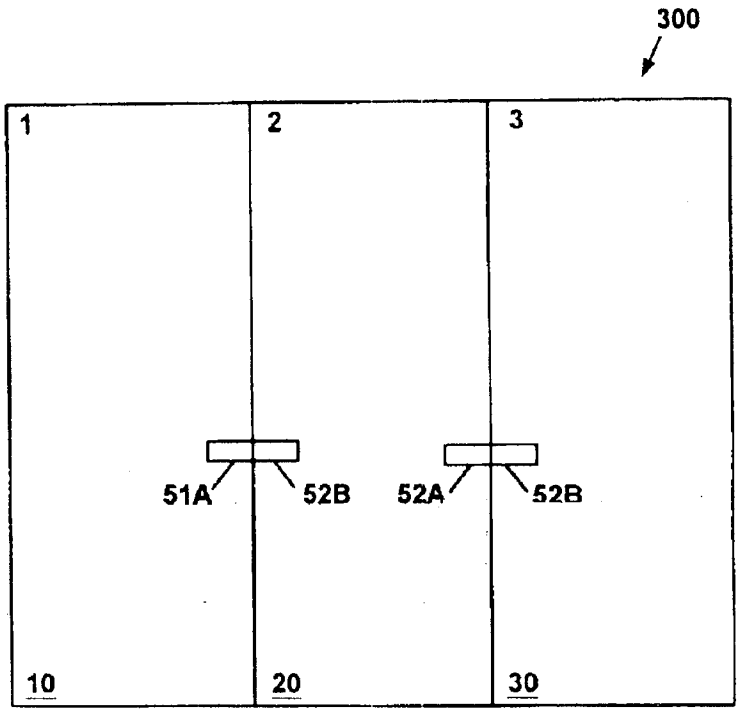


FIGURE 14B

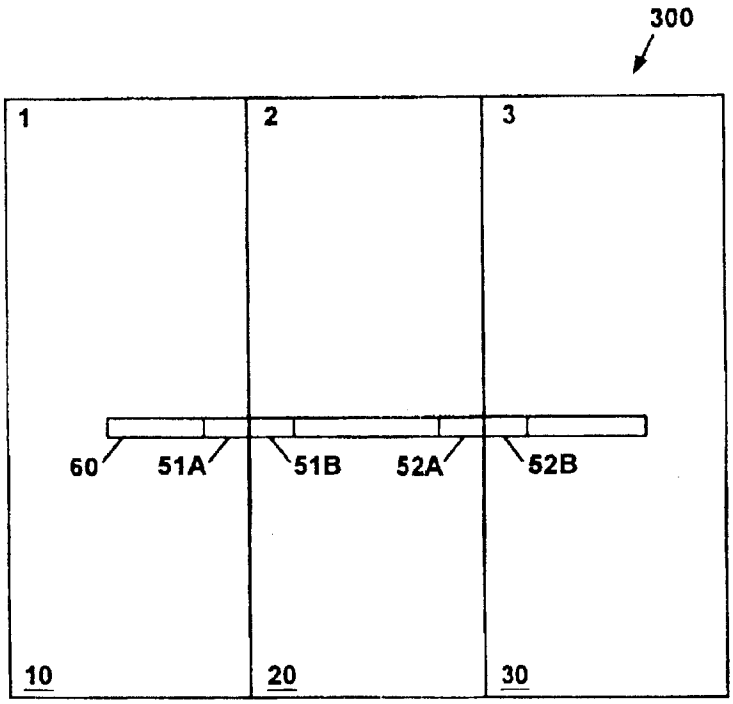


FIGURE 14C

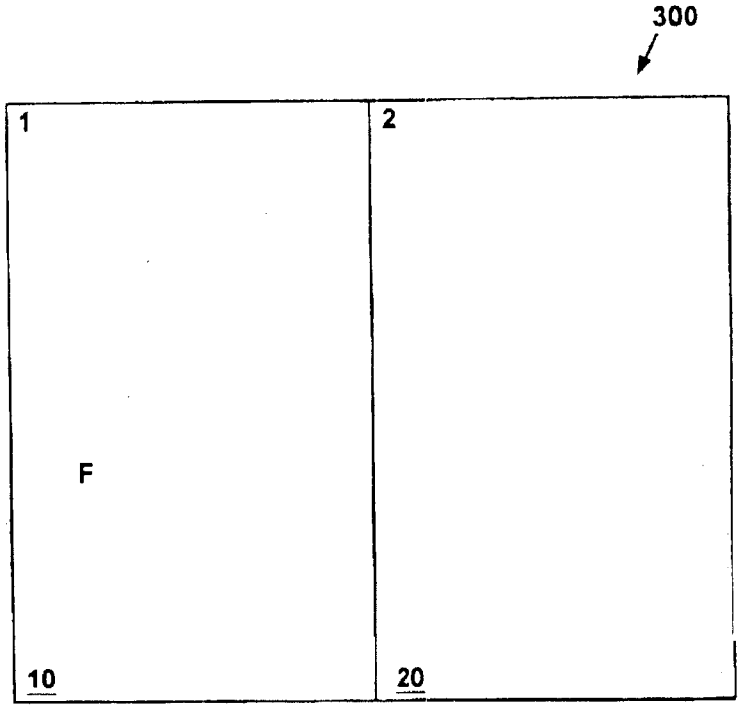


FIGURE 15A

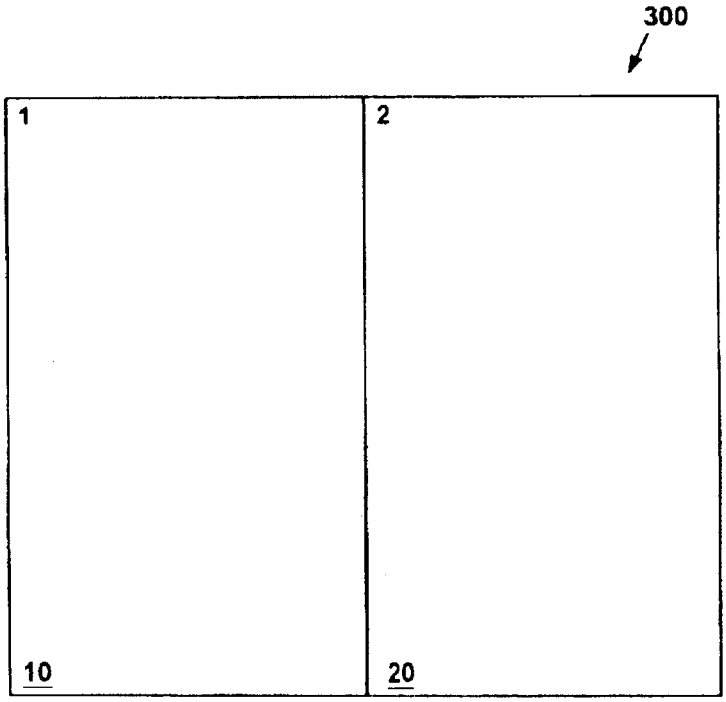


FIGURE 15B

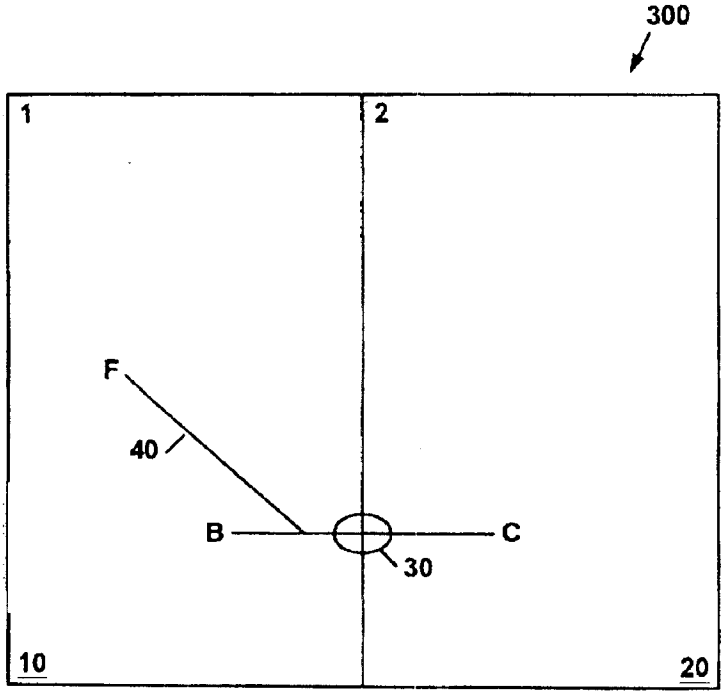


FIGURE 16A

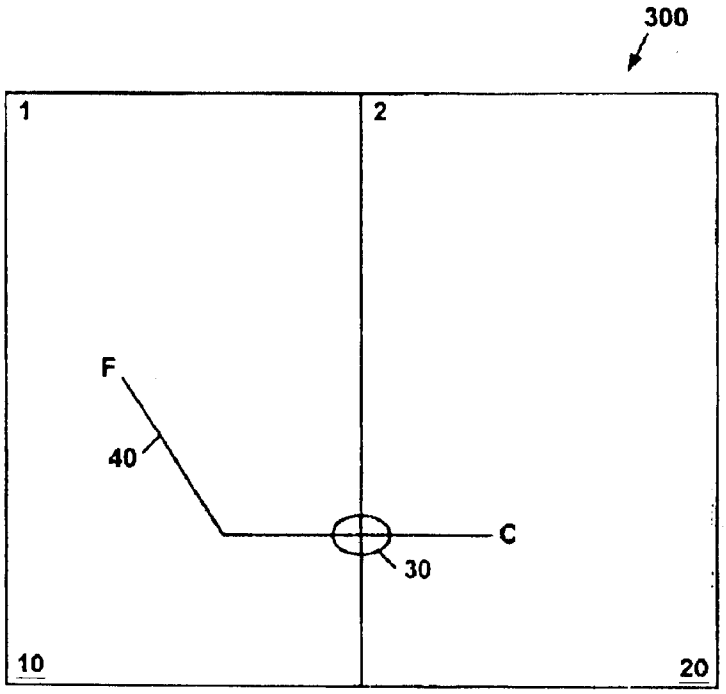


FIGURE 16B

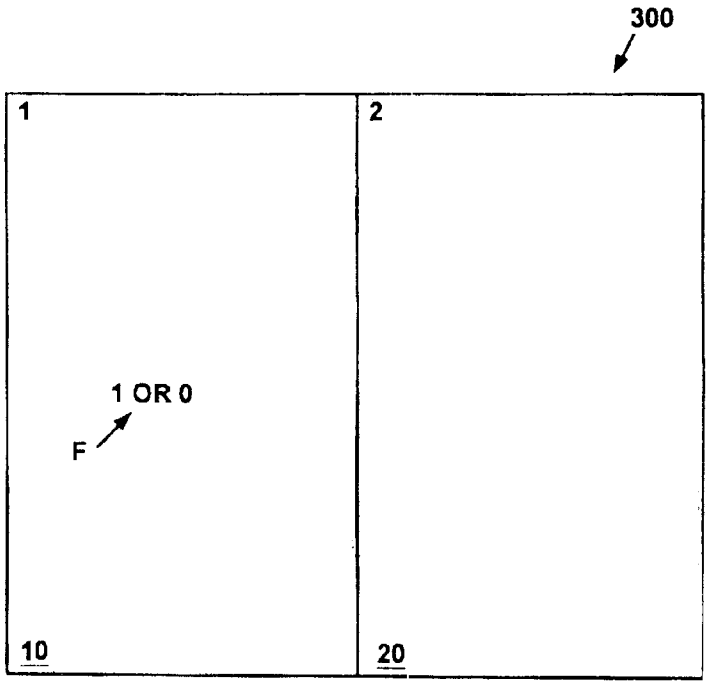


FIGURE 17A

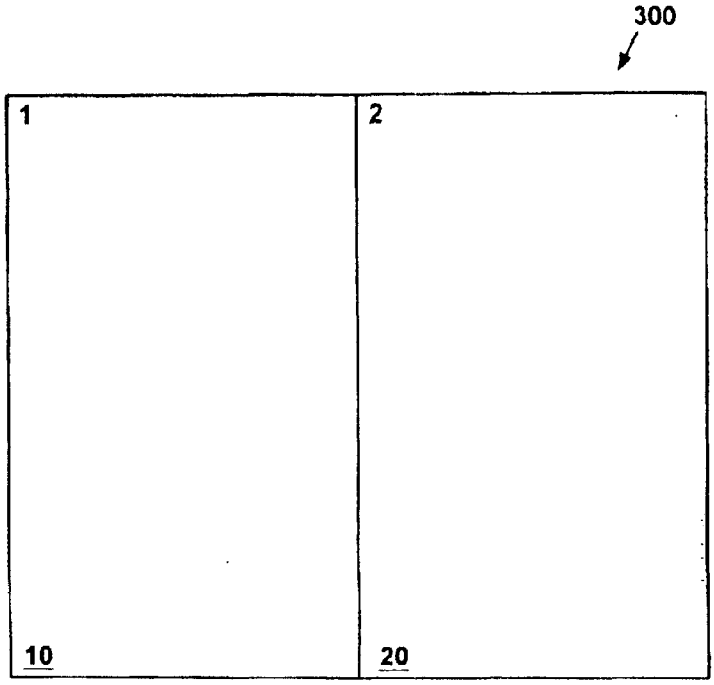


FIGURE 17B

FACILITATING PRESS OPERATION IN ABUTTED-PIN HIERARCHICAL PHYSICAL DESIGN

This patent application is a continuation of application Ser. No. 09/714,722, filed Nov. 15, 2000, entitled "OPTIMIZATION OF ABUTTED-PIN HIERARCHICAL PHYSICAL DESIGN", by Dahl et al., which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to the field of integrated circuit design. More particularly, the present invention relates to the field of software tools for hierarchical physical design.

2. Related Art

The tremendous advances in technology have been fueled by improvements in integrated circuit design. In particular, integrated circuits have become smaller and more complex. Integrated circuit design engineers depend on electronic design automation (EDA) software tools to facilitate the design of integrated circuits.

Typically, the integrated circuit design process begins with a specification which describes the functionality of the integrated circuit and may include a variety of constraints. Then, during a logic design phase, the logical implementation of the integrated circuit is determined. Several operations are performed to obtain a logical representation of the integrated circuit. Generally, EDA software tools use register transfer logic (RTL) to represent the integrated circuit. However, additional EDA software tools may be used.

After completing the logic design phase, the integrated circuit undergoes a physical design phase. Typically, the output of the logic design phase is a netlist, which is then used in the physical design phase. Here, EDA software tools layout the integrated circuit to obtain a representation of the physical components in the integrated circuit, whereas the representation indicates the manner in which the integrated circuit will be implemented on a semiconductor chip. A variety of operations are performed on the layout of the integrated circuit.

At the end of the physical design phase, the representation of the semiconductor chip (in which the integrated circuit is implemented) is sent to a semiconductor manufacturing plant.

Typically, in the physical design phase, EDA software tools implement a flat physical design. For example, the components (standard cells, macrocells, etc.) of the integrated circuit are placed during a placement operation and are routed during a routing operation. However, as the integrated circuit becomes more complex, the EDA software tools struggle to perform the placement operation and the routing operation. In particular, the performance of the EDA software tools degrades since the EDA software tools have to manipulate very large files during the placement operation and the routing operation. Moreover, as the complexity of the integrated circuit increases, the time necessary to complete the physical design phase increases significantly.

Traditional hierarchical physical design has emerged as an alternative to the flat physical design. FIG. **1** illustrates the traditional hierarchical physical design **100**. Here, the components of the integrated circuit are partitioned into a plurality of blocks **10–30**. Each block **10–30** includes a plurality of pins **50**, whereas each pin **50** represents a location where a signal can enter the block **10–30** or a location where a signal can exit the block **10–30**. As illustrated in FIG. **1**, the traditional hierarchical physical design **100** includes a channel **40**. The channel **40** provides space in order to connect the pins **50** of the blocks **10–30** to one another via metal (not shown) or any other wiring material. The traditional hierarchical physical design **100** enables the placement operation and the routing operation (as well as other operations) for the blocks **10–30** to be performed in parallel with EDA software tools, reducing the time period of the physical design phase. Moreover, the performance of the EDA software tools is improved because the file for each block **10–30** is much smaller than the file for the entire integrated circuit of the flat physical design. More importantly, the EDA software tools are better suited to optimize each block **10–30** than to optimize the entire integrated circuit of the flat physical design. However, the traditional hierarchical physical design **100** generates wasted space in the channel **40** and generates wiring problems in the channel **40**, such as congestion and crosstalk. Moreover, the traditional hierarchical physical design **100** places and routes components at a top-level (shown in FIG. **1**) and a block-level (within each block **10–30**), causing inefficiencies and causing problems with EDA software tools which are configured to operate with flat physical designs.

SUMMARY OF THE INVENTION

An abutted-pin hierarchical physical design process is described. The abutted-pin hierarchical physical design provides solutions to the problems of the traditional hierarchical physical design and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced significantly.

In the integrated circuit design flow according to an embodiment of the present invention, the physical design phase receives the netlist from the logic design phase. In addition, the physical design phase receives physical design information, whereas the physical design information can be any information about a prior integrated circuit that has undergone the physical design phase. In an embodiment, the physical design information is stored in a database.

In an embodiment of the present invention, the integrated circuit design flow of the present invention is utilized to optimize pin assignment. In an embodiment of the present invention, excess pins formed along a boundary between two blocks are removed.

In an embodiment of the present invention, a software tool that performs a "press" operation preserves the properties associated with a segment of a top-level shape despite the shape operation (e.g., AND) being performed with the block and the top-level shape to obtain the segment.

If the top-level object has the press property, the top-level object retains its location when the top-level object is "pressed" into a block. If the top-level object does not have the press property, the top-level object generally does not retain its location when the top-level object is "pressed" into the block.

If in the top-level netlist, the instantiation of a block includes a port that is unused, (thus, not needed for the top-level routing for pin assignment), a software tool removes the port from the top-level netlist, but the block-level netlist of the block remains unchanged.

Some software tools are not able to represent the relationship that more than one port is coupled to a pin. Hence, a software tool removes one of the ports from the netlist based on some criteria, such as whether a port is an input port or an output port.

If in the top-level netlist, the instantiation of the block includes a port that is tied to either the power line (1) or the ground line (0) rather to a port of another block, a software tool removes the port from the top-level netlist to avoid routing the port at the top-level. Moreover, the software tool ties the port to either the power line (1) or the ground line (0) in the block-level netlist of the block.

In an embodiment, a software tool performs an unwinding operation which adds to the block-level netlist—of bonding pad blocks—the ports (which were removed earlier by the software tool) that couple to the top-level inputs and to the top-level outputs. Thus, the netlist modified by the physical design phase (e.g., repeater and buffers are added to the netlist) can be compared with the netlist originally received from the logic design phase. In particular, formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification can be performed by software tools.

In an embodiment, each block-level netlist is partitioned into a first netlist and a second netlist. The second netlist and its associated extraction file of each block and the top-level netlist and its associated extraction file are utilized by software tools to perform the timing analysis. This timing analysis can be performed significantly faster than the case where the block-level netlist is not partitioned into the first netlist and the second netlist. In an embodiment, the timing graph resulting from the timing analysis can be analyzed to extract timing constraints (relating to the delay that can be generated by a block) for each block. Hence, if a block is optimized to meet its extracted timing constrains, the block is more likely to meet its timing parameter when the block interacts with the other blocks in the integrated circuit.

These and other advantages of the present invention will no doubt become apparent to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the present invention.

FIG. **1** illustrates the traditional hierarchical physical design **100**.

FIG. **2** illustrates an exemplary computer system **200** on which embodiments of the present invention may be practiced.

FIG. **3** illustrates an integrated circuit **300** generated with software tools according to an embodiment of the abutted-pin hierarchical physical design process of the present invention.

FIG. **4** illustrates the abutted-pin hierarchical physical design process **400** according to an embodiment of the present invention.

FIG. **5** illustrates the abutted-pin hierarchical physical design process **500** as performed at the block-level in a particular block (**450**A–**450**C of FIG. **4**) after step **440** of FIG. **4**.

FIG. **6** illustrates the layout of the blocks **10**–**30** is established.

FIG. **7** illustrates a clock wire **320** and a power wire **310** of the top-level.

FIG. **8** illustrates a top-level route for obtaining the pin assignments for each block **10**–**30**.

FIG. **9**A illustrates the integrated circuit design flow of the prior art.

FIG. **9**B illustrates the integrated circuit design flow according to an embodiment of the present invention.

FIG. **10**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the prior art (FIG. **9**A), showing the top-level routing for pin assignment.

FIG. **10**B illustrates the integrated circuit **300** of FIG. **10**A at the block level.

FIG. **10**C illustrates the integrated circuit **300** of FIG. **10**B at the block-level.

FIG. **11**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the present invention (FIG. **9**B), showing the top-level routing for pin assignment.

FIG. **11**B illustrates the integrated circuit **300** of FIG. **11**A at the block-level.

FIG. **11**C illustrates the integrated circuit **300** of FIG. **11**B at the block-level.

FIG. **12**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **12**B illustrates the integrated circuit **300** of FIG. **12**A at the block-level.

FIG. **12**C illustrates the integrated circuit **300** of FIG. **12**B, showing the removal of excess pins.

FIG. **13**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **30** (e.g., routing metal).

FIG. **13**B illustrates the segment **30**A of FIG. **13**A.

FIG. **13**C illustrates the integrated circuit **300** of FIG. **13**A in the top-level, showing that the segment **30**A has been removed from the top-level netlist and merged into the block-level netlist of block**1** **10**.

FIG. **14**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **60** (e.g., routing metal).

FIG. **14**B illustrates the integrated circuit **300** at the block-level.

FIG. **14**C illustrates the integrated circuit **300** at the block-level.

FIG. **15**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **15**B illustrates that the port F of block**1** **10** has been removed from the top-level netlist.

FIG. **16**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **16**B illustrates that the port B of block**1** **10** has been removed from the netlist for the top-level routing for pin assignment.

FIG. **17**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment.

FIG. **17**B illustrates that the port F of block**1 10** has been removed from the top-level netlist.

The drawings referred to in this description should not be understood as being drawn to scale except if specifically noted.

DETAILED DESCRIPTION OF THE INVENTION

Reference will now be made in detail to the preferred embodiments of the present invention, examples of which are illustrated in the accompanying drawings. While the invention will be described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be recognized by one of ordinary skill in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

Notation and Nomenclature

Some portions of the detailed descriptions which follow are presented in terms of procedures, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. In the present application, a procedure, logic block, process, etc., is conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proved convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, a variety of terms are discussed that refer to the actions and processes of an electronic system or a computer system, or other electronic computing device/system. The computer system or similar electronic computing device manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices. The present invention is also well suited to the use of other computer systems such as, for example, optical, mechanical, or quantum computers.

Exemplary Computer System Environment

Aspects of the present invention are discussed in terms of steps executed on a computer system. Although a variety of different computer systems can be used with the present invention, an exemplary computer system **200** is shown in FIG. **2**.

With reference to FIG. **2**, portions of the present invention are comprised of computer-readable and computer executable instructions which reside, for example, in computer-usable media of an electronic system such as the exemplary computer system. FIG. **2** illustrates an exemplary computer system **200** on which embodiments of the present invention may be practiced. It is appreciated that the computer system **200** of FIG. **2** is exemplary only and that the present invention can operate within a number of different computer systems including general-purpose computer systems and embedded computer systems.

Computer system **200** includes an address/data bus **110** for communicating information, a central processor **101** coupled with bus **110** for processing information and instructions, a volatile memory **102** (e.g., random access memory RAM) coupled with the bus **110** for storing information and instructions for the central processor **101** and a non-volatile memory **103** (e.g., read only memory ROM) coupled with the bus **110** for storing static information and instructions for the processor **101**. Exemplary computer system **200** also includes a data storage device **104** ("disk subsystem") such as a magnetic or optical disk and disk drive coupled with the bus **110** for storing information and instructions. Data storage device **104** can include one or more removable magnetic or optical storage media (e.g., diskettes, tapes) which are computer readable memories. Memory units of computer system **200** include volatile memory **102**, non-volatile memory **103** and data storage device **104**.

Exemplary computer system **200** can further include an optional signal generating device **1108** (e.g., a network interface card "NIC") coupled to the bus **1110** for interfacing with other computer systems. Also included in exemplary computer system **200** of FIG. **2** is an optional alphanumeric input device **106** including alphanumeric and function keys coupled to the bus **110** for communicating information and command selections to the central processor **101**. Exemplary computer system **200** also includes an optional cursor control or directing device **107** coupled to the bus **110** for communicating user input information and command selections to the central processor **101**. An optional display device **105** can also be coupled to the bus **110** for displaying information to the computer user. Display device **105** may be a liquid crystal device, other flat panel display, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters recognizable to the user. Cursor control device **107** allows the user to dynamically signal the two-dimensional movement of a visible symbol (cursor) on a display screen of display device **105**. Many implementations of cursor control device **107** are known in the art including a trackball, mouse, touch pad, joystick or special keys on alphanumeric input device **106** capable of signaling movement of a given direction or manner of displacement. Alternatively, it will be appreciated that a cursor can be directed and/or activated via input from alphanumeric input device **106** using special keys and key sequence commands.

Abutted-Pin Hierarchical Physical Design

FIG. **3** illustrates an integrated circuit **300** generated with software tools according to the abutted-pin hierarchical physical design process of the present invention. The abutted-pin hierarchical physical design provides solutions to the problems of the traditional hierarchical physical design (see FIG. **1**) and provides additional advantages and benefits. In particular, the abutted-pin hierarchical physical design does not have channels. Moreover, in the abutted-pin hierarchical physical design, components of the top-level are merged into the block-level so that the top-level netlist is reduced to instantiations of each block **10–30** and **60–94**.

As illustrated in FIG. **3**, the abutted-pin hierarchical physical design **300** includes a plurality of blocks **10–30** and **60–94**. The netlist of the integrated circuit **300** is partitioned into the plurality of blocks **10–30** and **60–94** such that each block **10–30** and **60–94** has a block level netlist. Blocks **10–30** have the major or core components of the integrated circuit **300**. Blocks **60–94** have the bonding pads and other support circuitry of the integrated circuit **300**. The blocks **10–30** and **60–94** can be rectangular in shape and can be rectilinear in shape. It should be understood that the integrated circuit **300** can have any number of blocks.

Each block **10–30** and **60–94** has one or more pins **50**, whereas each pin **50** represents a location where a signal can enter the block **10–30** and **60–94** or a location where a signal can exit the block **10–30** and **60–94**. The edge or boundary of each block **10–30** and **60–94** rests against the edge or boundary of another block **10–30** and **60–94**, such that the pin **50** of one block abuts the pin **50** of another block.

Moreover, the top-level components or objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) are not visible because they have been merged into the blocks **10–30** and **60–94** by a "press" operation performed by a software tool. First, the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) are placed and routed at the top-level (the top-level is shown in FIG. **3**). In the "press" operation, the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.) that are within the boundary of a block **10–30** and **60–94** are removed from the top-level netlist and merged into the block-level netlist of that block **10–30** and **60–94**. Hence, the abutted-pin hierarchical physical design **300** can be optimized by separately optimizing the individual blocks **10–30** and **60–94**. Thus, the software tools can generate (e.g., perform placement, routing, timing, verification, etc.) and optimize the individual blocks **10–30** and **60–94** in parallel. Moreover, a bug within an individual block **10–30** and **60–94** can be corrected by returning that individual block to the logic design phase, while the other blocks continue to undergo the physical design phase.

FIG. **4** illustrates the abutted pin hierarchical physical design process **400** according to an embodiment of the present invention. At **410**, a software tool receives the netlist of the integrated circuit from the logic design phase, as described above. The netlist is partitioned into a plurality of blocks, each block having a block-level netlist. In an embodiment, the partitioning of the netlist focuses on reducing the number of ports or terminals of a block that need to couple to the ports or terminals of other blocks.

At **420**, a software tool performs top-level floor planning. Here, the layout of each block is determined. At the end of the top-level floor planning, the top-level for an integrated circuit **300** (as shown in FIG. **6**) is generated. As illustrated in FIG. **6**, the layout of the blocks **10–30** is established. In FIG. **6**, the bonding pads **60–94** (of FIG. **3**) have been omitted.

At **430**, software tools perform top-level placement and routing for the top-level objects (e.g., timing components, clock distribution wiring, power distribution wiring, repeaters, buffers, etc.). FIG. **7** illustrates a clock wire **320** and a power wire **310** of the top-level. The clock wire **320** is routed over BlockA **10** and BlockC **30**. The power wire **310** is routed over BlockA **10**. It should be understood that any number of additional top-level objects can be placed and routed at the top-level.

At **440**, a software tool performs a top-level route for obtaining the pin assignments for each block **10–30**, as illustrated in FIG. **8**. Since each block **10–30** has one or more ports or terminals **47** that needs to couple to a port or terminal of another block **10–30**, the pins for each block **10–30** have to be defined. Initially, the ports **47** of each block **10–30** are placed in a general random location within each block at the top-level since the actual location of the port **47** is not known until a placement operation is performed at the block-level. As illustrated in FIG. **8**, the location **45A–45F** where a routing wire **48** crosses a boundary between two blocks is defined as a pin for each of the blocks **10–30**, facilitating creation of pins that are abutted. In an embodiment, a software tool creates each pin to have a width that is equivalent to the width of the routing wire **48** at the boundary between the two blocks. The pins **50** are illustrated in FIG. **3**.

At **450A–450C**, the abutted-pin hierarchical physical design process **400** enables software tools to generate and to optimize each block **10–30** in parallel at the block-level.

FIG. **5** illustrates the abutted-pin hierarchical physical design process **500** as performed at the block-level in a particular block (**450A–450C** of FIG. **4**) after step **440** of FIG. **4**.

At **510**, a software tool performs press operations. The top-level objects illustrated in FIG. **7** (e.g., a clock wire **320** and a power wire **310**) and which are located within the boundary of a particular block, are pressed into the particular block. In particular, the top-level objects that are within the boundary of a particular block are removed from the top-level netlist and merged into the block-level netlist of that particular block. Moreover, the pins for the particular block are generated based on the location where the routing wire crosses the boundary between two blocks, as illustrated in FIG. **8** and FIG. **3**.

At **520**, a software tool performs block-level floor planning for the particular block. At **530**, a software tool performs a block-level placement operation for the particular block. At **540**, software tools perform a variety of block-level operations to optimize the particular block. Additionally, at **550**, a block-level route is performed for the particular block by a software tool. At **552** and **554**, software tools perform a block-level extraction operation for determining capacitance and resistance at the nodes and perform block-level timing analysis operations for the particular block.

At **560** and **570**, a variety of software tools perform a number of verification operations such as formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification.

FIG. **9A** illustrates the integrated circuit design flow of the prior art. As illustrated in FIG. **9A**, the physical design phase **910** receives the netlist from the logic design phase (not shown). The physical design phase **910** generates the physi-

cal design for the integrated circuit and outputs a GDS II file. The GDS II file is received by the semiconductor factory **920**. The integrated circuit is fabricated by the semiconductor factory **920** on a semiconductor chip.

FIG. **9**B illustrates the integrated circuit design flow according to an embodiment of the present invention. As illustrated in FIG. **9**B, the physical design phase **910** receives the netlist from the logic design phase (not shown). In addition, the physical design phase **910** receives physical design information **930**, whereas the physical design information **930** can be any information about a prior integrated circuit that has undergone the physical design phase **910**. In an embodiment, the physical design information **930** is stored in a database. For example, the physical design information **930** can be pin assignments of the prior integrated circuit, optimal clock distribution tree of the prior integrated circuit, parasitic extraction data of the prior integrated circuit, locations of obstructions such as a RAM of the prior integrated circuit, identification of congested blocks of the prior integrated circuit, metal resources for the blocks of the prior integrated circuit, or any other information which can facilitate optimizing the current integrated circuit. Thus, the software tools of the physical design phase **910** can customize the current integrated circuit to avoid the problems of the prior integrated circuit and to realize the benefits of the prior integrated circuit.

In the physical design phase **910**, decisions made at the top-level with respect to the top-level objects, significantly influence the creation of problems at the block-level and the optimization operations at the block-level. By using physical design information **930** (concerning the block-level of the prior integrated circuit) at the top-level of the current integrated circuit, the decisions made at the top-level with respect to the top-level objects of the current integrated circuit will be able to reduce the problems present in the prior integrated circuit and will be able to generate solutions to overcome the problems present in the prior integrated circuit, improving the optimization of the abutted-pin hierarchical physical design process of the present invention. Thus, if the physical design information **930** has information about several prior integrated circuits, the current integrated circuit is more likely to be optimized.

In addition, the physical design phase **910** generates the physical design for the integrated circuit and outputs a GDS II file. Moreover, the physical design phase **910** stores physical design information **930** of the current integrated circuit to be used in the physical design phase **910** of a future integrated circuit. The GDS II file is received by the semiconductor factory **920**. The integrated circuit is fabricated by the semiconductor factory **920** on a semiconductor chip.

FIG. **10**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the prior art (FIG. **9**A), showing the top-level routing for pin assignment. The port C of block1 **10** is routed to port B of block2 **20**. The port A of block1 **10** is routed to port D of block **20**. This top-level routing has been performed after ports A–D where placed in a generally random location within each block **10**–**20** at the top-level since the actual locations of the ports A–D are not known until a placement operation is performed at the block-level. Here, the software tools at the top-level do not have access to the physical design information of a prior integrated circuit. The locations **15** and **16** are where the routing metal **18** crosses the boundary between two blocks **10** and **20**.

FIG. **10**B illustrates the integrated circuit **300** of FIG. **10**A. At the block-level, the pins **15**A and **16**A were formed for block1 **10**. At the block-level, the pins **15**B and **16**B were formed for block2 **20**, whereas pin **15**A abuts pin **15**B and pin **16**A abuts pin **16**B. The pins **15**A and **15**B were formed at location **15** of FIG. **10**A. The pins **16**A and **16**B were formed at location **16** of FIG. **10**A.

FIG. **10**C illustrates the integrated circuit **300** of FIG. **10**B at the block-level. As illustrated in FIG. **10**C, the block-level placement operation for block1 **10** placed the ports A and C at locations that are different from the locations used to generate the pin assignments in FIG. **10**A. In addition, the block-level placement operation for block2 **20** placed the ports B and D at locations that are different from the locations used to generate the pin assignments in FIG. **10**A. Hence, the block-level routing operations for blocks **10** and **20** generated an inefficient amount of routing wire **19** to couple the ports to the pins in each block. In sum, the pin assignment affects the optimization of the routing wire **19**.

FIG. **11**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention and using the integrated circuit design flow of the present invention (FIG. **9**B), showing the top-level routing for pin assignment. The port C of block1 **10** is routed to port B of block2 **20**. The port A of block1 **10** is routed to port D of block2 **20**. This top-level routing has been performed after each port A–D where placed in a particular location within each block **10**–**20** at the top-level, whereas the particular location was based on using the physical design information associated with the prior integrated circuit (FIGS. **1**A–**10**C). Here, the software tools at the top-level have access to the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C). The locations **15** and **16** are where the routing metal **18** crosses the boundary between two blocks **10** and **20**.

FIG. **11**B illustrates the integrated circuit **300** of FIG. **11**A at the block-level. At the block-level, the pins **15**A and **16**A were formed for block1 **10**. At the block-level, the pins **15**B and **16**B were formed for block2 **20**, whereas pin **15**A abuts pin **15**B and pin **16**A abuts pin **16**B. The pins **15**A and **15**B were formed at location **15** of FIG. **11**A. The pins **16**A and **16**B were formed at location **16** of FIG. **11**A. Here, the pins **15**A and **15**B are associated with ports A and D, unlike FIG. **10**B where pins **15**A and **15**B were associated with ports C and B. Moreover, the pins **16**A and **16**B of FIG. **11**B are associated with ports C and B, unlike FIG. **10**B where pins **16**A and **16**B were associated with ports A and D.

FIG. **11**C illustrates the integrated circuit **300** of FIG. **11**B at the block-level. As illustrated in FIG. **11**C, the block-level placement operation for block1 **10** placed the ports A and C at locations that are different from the locations used to generate the pin assignments in FIG. **11**A. In addition, the block-level placement operation for block2 **20** placed the ports B and D at locations that are different from the locations used to generate the pin assignments in FIG. **11**A. However, the difference in the location of the ports between FIG. **11**A and FIG. **11**C is less than the difference in the location of the ports between FIG. **10**A and FIG. **10**C. Hence, the block-level routing operations for blocks **10** and **20** generated a more efficient amount of routing wire **19** to couple the ports to the pins in each block, compared to FIG. **10**C. In sum, the pin assignments generated with the use of the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C) were more optimal than the pin assignments generated without the use of the physical design information of the prior integrated circuit (FIGS. **10**A–**10**C).

FIG. **12**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the

present invention, showing the top-level routing for pin assignment. In the course of routing source port **24** of block3 **30** to destination port **22** of block2 **20**, the software tool that performs the top-level routing for pin assignment crosses the boundary between block1 **10** and block2 **20** at locations **15**,**16**, and **17**, whereas the locations **15**,**16**, and **17** will be defined as pins. The software tool is concerned with routing a path between the source port **24** and the destination port **22**, but is not concerned about the number of times the path crosses the boundary between the same blocks.

FIG. **12**B illustrates the integrated circuit **300** of FIG. **12**A at the block-level. The pins **15**A–**15**B, **16**A–**16**B, and **17**A–**17**B are formed between block1 **10** and block2 **20**. The pins **18**A–**18**B are formed between block1 **10** and block3 **30**. The presence of pins **16**A–**16**B and **17**A–**17**B causes additional routing metal to be added to block1 **10** and block2 **20** so that pins **15**A, **16**A, and **17**A can be coupled within block1 **10** and so that pins **15**B, **16**B, and **17**B can be coupled within block2 **20**. Hence, one pair of pins (**15**A–**15**B or **16**A–**16**B or **17**A–**17**B) is sufficient.

FIG. **12**C illustrates the integrated circuit **300** of FIG. **12**B, showing the removal of excess pins. As illustrated in FIG. **12**C, excess pins **16**A–**16**B and **17**A–**17**B were removed from block1 **10** and block2 **20**. This removal is based on a plurality of criteria, such as the current flow direction between the source port **24** and the destination port **22**, the location of the excess pins relative to the source port **24** and the destination port **22**, or any other criteria. Here, the criteria kept pins **15**A–**15**B but deleted pins **16**A–**16**B and **17**A–**17**B.

FIG. **13**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **30** (e.g., routing metal). As described above, a software tool performs a press operation so that the portion of the top-level object **30** which is within the boundary of a particular block **10**–**20** is moved from the top-level netlist to the block-level netlist of the particular block **10**–**20**. In particular, the segment **30**A is pressed into block1 **10** while the segment **30**B is pressed into block2 **20**. In an embodiment, the shape operations of a database are utilized in performing the press operation. In FIG. **13**A, an AND operation would be performed with block1 **10** and the shape **30** to obtain the segment **30**A (FIG. **13**B). Typically, the routing metal **30** includes a plurality of properties that are stored in a database. These properties identify the routing metal **30** and describe the function of the routing metal **30**. However, in the shape operations (e.g., AND) of the prior art, the shape operation returns the segment **30**A (FIG. **13**B) without its properties. Thus, these properties have to be reconstructed.

In the present invention, the software tool that performs the press operation preserves the properties associated with segment **30**A of the routing metal **30** despite the shape operation (e.g., AND) performed with block1 **10** and the shape **30** to obtain the segment **30**A (FIG. **13**B).

FIG. **13**C illustrates the integrated circuit **300** of FIG. **13**A in the top-level, showing that the segment **30**A has been removed from the top-level netlist and merged into the block-level netlist of block1 **10**. Moreover, the properties associated with segment **30**A at the top-level are transferred to the segment **30**A at the block-level.

FIG. **14**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for a top-level object **60** (e.g., routing metal). As illustrated in FIG. **14**A, the top-level object **60** is routed through block1 **10**, block2 **20**, and block3 **30**. The locations **51**–**52** indicate top-level object **60** crosses a boundary between two blocks. In an embodiment, a press property is added to the properties of the top-level object **60** stored in a database. If the top-level object **60** has the press property, the top-level object **60** retains its location when the top-level object **60** is pressed into block1 **10**, block2 **20**, and block3 **30**, as illustrated in the block-level view of the integrated circuit **300** in FIG. **14**C. If the top-level object **60** does not have the press property, the top-level object **60** generally does not retain its location when the top-level object **60** is pressed into block1 **10**, block2 **20**, and block3 **30**, as illustrated in the block-level view of the integrated circuit **300** in FIG. **14**B. For example, top-level objects such as power and ground have the press property. As illustrated in FIG. **14**B, the pins **51**A–**51**B and **52**A–**52**B are defined. However, the software tool is not constrained to placing the top-level object **60** in the block-level exactly as it was placed at the top-level. Moreover, the top-level object is placed in the block-level of block1 **10**, block2 **20**, and block3 **30** according to the separate placement and routing requirements of block1 **10**, block2 **20**, and block3 **30**.

FIG. **15**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **15**A, in the top-level netlist, the instantiation of block1 **10** includes a port F that is unused, thus, not needed for the top-level routing for pin assignment. Hence, a software tool removes port F from the top-level netlist, but the block-level netlist of block1 **10** remains unchanged. In an embodiment, the software tool that performs the press operation removes the port F. FIG. **15**B illustrates that the port F of block1 **10** has been removed from the top-level netlist.

FIG. **16**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **16**A, port F and port B of block1 **10** are coupled to port C of block2 **20** with a routing metal **40**. However, at location **30** the routing metal **40** crosses the boundary between block1 **10** and block2 **20**. If a pin is formed within block1 **10** at location **30**, the pin would be coupled to port F and to port B. However, some software tools are not able to represent this relationship (i.e., more than one port coupled to a pin). Hence, a software tool removes one of the ports (port F or port B) from the netlist based on some criteria, such as whether a port is an input port or an output port. FIG. **16**B illustrates that the port B of block1 **10** has been removed from the netlist for the top-level routing for pin assignment.

FIG. **17**A illustrates an integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention, showing the top-level routing for pin assignment. As illustrated in FIG. **17**A, in the top-level netlist, the instantiation of block1 **10** includes a port F that is tied to either the power line (1) or the ground line (0) rather to a port of another block. Hence, a software tool removes port F from the top-level netlist to avoid routing the port F at the top-level. Moreover, the software tool ties the port F to either power line (1) or the ground line (0) in the block-level netlist of block1 **10**. FIG. **17**B illustrates that the port F of block1 **10** has been removed from the top-level netlist.

As illustrated in FIG. **3**, the integrated circuit **300** based on the abutted-pin hierarchical physical design process of the present invention includes a North bond pad block **60**, an

East bond pad block **70**, a South bond pad block **80**, and a West bond pad block **90**, each having bond pad cells. The top-level netlist of the integrated circuit **300** includes one or more top-level inputs for receiving external signals and one or more top-level outputs for transmitting signals off the chip. The top-level inputs and the top-level outputs are coupled to bond pad cells. Typically, software tools which perform a routing operation are configured to not perform the routing operation if the netlist includes bond pad cells. Since the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** have bond pad cells in the block-level netlist, the software tools refuse to perform the routing operation in these blocks, preventing pins to be formed on the boundary between these blocks and the blocks **10–30** (the core blocks).

In the present invention, the bond pad cells are marked as macrocells rather than bond pad cells, allowing pins to be formed on the boundary between these blocks **60**, **70**, **80**, and **90** and the blocks **10–30** (the core blocks).

Typically, the block-level netlist of the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** include nets to the top-level inputs and nets to the top-level outputs. Generally, the block-level-netlist of the North bond pad block **60**, the East bond pad block **70**, the South bond pad block **80**, and the West bond pad block **90** include nets to the bond pad cells.

In an embodiment of the present invention, a software tool removes the nets to the top-level inputs and nets to the top-level outputs so that the physical design of the integrated circuit can be accomplished as described above. In an embodiment, the software tool removes in the block-level netlist the ports that couple to the top-level inputs and to the top-level outputs. Moreover, the software tool adds a property to the nets to the bond pad cells to indicate that these nets are suppose to couple to the top-level inputs and to the top-level outputs, facilitating an unwinding operation to rep establish at the block-level netlist the nets to the top-level inputs and nets to the top-level outputs that were removed earlier. The unwinding operation adds to the block-level netlist the ports (which were removed earlier) that couple to the top-level inputs and to the top-level outputs. Thus, the netlist modified by the physical design phase (e.g., repeater and buffers are added to the netlist) can be compared with the netlist originally received from the logic design phase. In particular, formal verification, layout versus schematic (LVS) verification, and design rules check (DRC) verification can be performed by software tools.

A challenge with implementing an integrated circuit based on the abutted-pin hierarchical physical design process of the present invention involves analyzing the timing of signal paths that traverse more than one block. The timing of these global paths is difficult to analyzed compared to analyzing the timing of local paths, whereas local paths are signal paths that do not leave a block. One method of analyzing the timing of these global paths involves partitioning the block-level netlist of each block into a first netlist and a second netlist. The first netlist includes nets which start at a register (or flip-flop) and end at a register (or flip-flop) within the block, whereas each branch of the net also starts at a register (or flip-flop) and ends at a register (or flip-flop) within the block. The second netlist includes nets which are coupled to a pin of the block. Generally, the first netlist is ¾ of the initial block-level netlist while the second netlist is ¼ of the initial block-level netlist. If the second netlist ratio is greater than ¼, this indicates inefficient partitioning of the blocks.

Once the first netlist and the second netlist are obtain, an extraction operation to obtain parasitic resistance and capacitance is performed on the second netlist of each block. In an embodiment, the partitioning of the block-level netlist and the extraction operation in each block are performed in parallel. Moreover, an extraction operation is performed on the top-level netlist. In an embodiment, a software tool replaces the abutted pins of the top-level netlist with zero ohm resistors.

Some software tools utilized to perform the timing analysis are unable to operate on netlists having nets that are coupled to multiple pins of a block. In an embodiment of the present invention, these netlist are transformed by using "assign statements" to assign different names to the nets that are coupled to multiple pins of a block. Hence, each different named net can be coupled to a separate pin of the block.

In an embodiment, the second netlist and its associated extraction file of each block and the top-level netlist and its associated extraction file are utilized by software tools to perform the timing analysis. This timing analysis can be performed significantly faster than the case where the block-level netlist is not partitioned into the first netlist and the second netlist. In an embodiment, the timing graph resulting from the timing analysis can be analyzed to extract timing constraints (relating to the delay that can be generated by a block) for each block. Hence, if a block is optimized to meet its extracted timing constrains, the block is more likely to meet its timing parameter when the block interacts with the other blocks in the integrated circuit.

The foregoing descriptions of specific embodiments of the present invention have been presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, and obviously many modifications and variations are possible in light of the above teaching. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims appended hereto and their equivalents.

What is claimed is:

1. A method of pressing a top-level object into one or more blocks of a physical design, comprising:

a) performing a shape operation to identify a portion of said top-level object that is within a boundary of a particular block, wherein said top-level object includes a plurality of properties, and wherein said shape operation preserves said properties associated with said portion of said top-level object; and

b) moving said portion of said top-level object and said properties associated with said portion of said top-level object from a top-level netlist to a block-level netlist of said particular block so that said top-level object changes from a flattened state to a hierarchical state.

2. A method as recited in claim **1** wherein said physical design is an abutted-pin hierarchical physical design.

3. A method as recited in claim **2** wherein said physical design includes a top-level physical design.

4. A method as recited in claim **2** wherein said physical design includes a block-level physical design.

5. A method as recited in claim **1** wherein said top-level object is a timing component.

6. A method as recited in claim **1** wherein said top-level object is a clock distribution wiring.

7. A method as recited in claim 1 wherein said top-level object is a power distribution wiring.

8. A method as recited in claim 1 wherein said top-level object is a routing metal.

9. A method as recited in claim 1 wherein said shape operation is an AND operation.

10. A method as recited in claim 1 wherein said properties are stored in a database.

11. A computer-readable medium comprising computer-executable instructions stored therein for performing a method of pressing a top-level object into one or more blocks of a physical design, said method comprising:

a) performing a shape operation to identify a portion of said top-level object that is within a boundary of a particular block, wherein said top-level object includes a plurality of properties, and wherein said shape operation preserves said properties associated with said portion of said top-level object; and

b) moving said portion of said top-level object and said properties associated with said portion of said top-level object from a top-level netlist to a block-level netlist of said particular block so that said top-level object changes from a flattened state to a hierarchical state.

12. A computer-readable medium as recited in claim 11 wherein said physical design is an abutted-pin hierarchical physical design.

13. A computer-readable medium as recited in claim 12 wherein said physical design includes a top-level physical design.

14. A computer-readable medium as recited in claim 12 wherein said physical design includes a block-level physical design.

15. A computer-readable medium as recited in claim 11 wherein said top-level object is a timing component.

16. A computer-readable medium as recited in claim 11 wherein said top-level object is a clock distribution wiring.

17. A computer-readable medium as recited in claim 11 wherein said top-level object is a power distribution wiring.

18. A computer-readable medium as recited in claim 11 wherein said top-level object is a routing metal.

19. A computer-readable medium as recited in claim 11 wherein said shape operation is an AND operation.

20. A computer-readable medium as recited in claim 11 wherein said properties are stored in a database.

21. A method of pressing a top-level object into one or more blocks of a physical design, comprising:

a) identifying a portion of said top-level object that is within a boundary of a particular block;

b) moving said portion of said top-level object from a top-level netlist to a block-level netlist of said particular block so that said top-level object changes from a flattened state to a hierarchical state;

c) if said top-level object includes a press property, performing a block-level placement for said particular block such that a block-level physical location of said portion of said top-level object is substantially equivalent to a top-level physical location of said portion of said top-level object; and

d) if said top-level object does not include a press property, performing said block-level placement for said particular block without regard to said top-level physical location of said portion of said top-level object.

22. A method as recited in claim 21 wherein said physical design is an abutted-pin hierarchical physical design.

23. A method as recited in claim 22 wherein said physical design includes a top-level physical design.

24. A method as recited in claim 22 wherein said physical design includes a block-level physical design.

25. A method as recited in claim 21 wherein said top-level object is a timing component.

26. A method as recited in claim 21 wherein said top-level object is a clock distribution wiring.

27. A method as recited in claim 21 wherein said top-level object is a power distribution wiring.

28. A method as recited in claim 21 wherein said top-level object is a routing metal.

29. A method as recited in claim 21 wherein said top-level object is a ground distribution wiring.

30. A method as recited in claim 21 wherein said press property is stored in a database.

31. A computer-readable medium comprising computer-executable instructions stored therein for performing a method of pressing a top-level object into one or more blocks of a physical design, said method comprising:

a) identifying a portion of said top-level object that is within a boundary of a particular block;

b) moving said portion of said top-level object from a top-level netlist to a block-level netlist of said particular block so that said top-level object changes from a flattened state to a hierarchical state;

c) if said top-level object includes a press property, performing a block-level placement for said particular block such that a block-level physical location of said portion of said top-level object is substantially equivalent to a top-level physical location of said portion of said top-level object; and

d) if said top-level object does not include a press property, performing said block-level placement for said particular block without regard to said top-level physical location of said portion of said top-level object.

32. A computer-readable medium as recited in claim 31 wherein said physical design is an abutted-pin hierarchical physical design.

33. A computer-readable medium as recited in claim 32 wherein said physical design includes a top-level physical design.

34. A computer-readable medium as recited in claim 32 wherein said physical design includes a block-level physical design.

35. A computer-readable medium as recited in claim 31 wherein said top-level object is a timing component.

36. A computer-readable medium as recited in claim 31 wherein said top-level object is a clock distribution wiring.

37. A computer-readable medium as recited in claim 31 wherein said top-level object is a power distribution wiring.

38. A computer-readable medium as recited in claim 31 wherein said top-level object is a routing metal.

39. A computer-readable medium as recited in claim 31 wherein said top-level object is a ground distribution wiring.

40. A computer-readable medium as recited in claim 31 wherein said press property is stored in a database.

* * * * *

EXHIBIT 12

FULLY REDACTED

EXHIBIT 13

FULLY REDACTED

EXHIBIT 14

Subject: Re: Dates?

From: Tamara Fraizer [mailto:Fraizer@fr.com]
Sent: Tuesday, February 27, 2007 5:20 PM
To: Wagner, Valerie; soliver@pooleyoliver.com
Cc: Skogstrom, Carol; Allison, Craig; Fisher, Michael; dtabesh@pooleyoliver.com; kolivares@pooleyoliver.com
Subject: RE: Dates?

Valerie,

Thanks for your email and for emailing us. We are indeed representing Mr. Rodman despite his email to your attorneys. Also, we did take action to confirm the date and were planning on this date as well. Unfortunately, as is clear from his email, Mr. Rodman is now not available.

In response to your question, we do not presently intend Mr. Rodman at trial. However, we are not in a position to guarantee that we will not call him. Please let me know how you'd like to proceed. I was in non-stop meetings today but am available to discuss this tomorrow if you would like.

Thanks,
Tamara

From: Wagner, Valerie [mailto:valerie.wagner@dechert.com]
Sent: Tuesday, February 27, 2007 12:19 PM
To: soliver@pooleyoliver.com; Tamara Fraizer
Cc: Skogstrom, Carol; Allison, Craig; Fisher, Michael; dtabesh@pooleyoliver.com; kolivares@pooleyoliver.com
Subject: RE: Dates?

Scott:

In light of Mr. Rodman's direct communication with our firm, please confirm that Magma's counsel is representing him. Also, I confirmed with Tamara on February 19 that we would proceed with Mr. Rodman's deposition on March 6. Obviously there was a delay of over a week in communicating this to him, so please also explain this as it is now creating a problem with scheduling.

Please also let me know whether Magma intends to call Mr. Rodman to testify at trial. If not and Magma will confirm this in writing, we can discuss better how to proceed in light of these scheduling difficulties. If you or Tamara wish to discuss this by phone, I am available most of the day.

Regards,

Valerie

From: Scott Oliver [mailto:soliver@pooleyoliver.com]
Sent: Tuesday, February 27, 2007 12:08 PM
To: Wagner, Valerie; Tamara Fraizer
Cc: Skogstrom, Carol; Allison, Craig; Fisher, Michael; Dara Tabesh; Kim Olivares; soliver@pooleyoliver.com
Subject: RE: Dates?

Valerie:

We received a similar message this morning. The parties delayed too long in agreeing on the offered date of March 6, and Mr. Rodman is now unavailable that entire week. He offers the

following dates:

- Tuesday, March 13 (I will be in Minnesota on another case and cannot be available)
- Wednesday, March 21 (I am available on that date)
- Thursday, March 22 (I am available on that date)
- Friday, March 23 (I am available on that date)

Please let us know which date Synopsys prefers as soon as possible. I am traveling and will be off email shortly, so please confirm with Tamara so that she can in turn confirm the date with Mr. Rodman.
Thanks,
--Scott

From: Wagner, Valerie [mailto:valerie.wagner@dechert.com]
Sent: Tuesday, February 27, 2007 11:21 AM
To: Scott Oliver; Tamara Fraizer
Cc: Skogstrom, Carol; Allison, Craig; Fisher, Michael
Subject: FW: Dates?

Scott and Tamara:
We received the attached email from Paul Rodman a few minutes ago, but are not responding as we understand that Magma's counsel is representing him. Please confirm that this is still the case and advise as to the scheduling of this deposition.
Regards,
Valerie

From: Paul Rodman [mailto:rodman@google.com]
Sent: Tuesday, February 27, 2007 2:01 PM
To: Fisher, Michael
Subject: Dates?

Scott Oliver seems to think I'm scheduled for depo on tuesday of next week.
I have recv'ed no email or vmail on any dates, and in fact that week is now totally gonzo, so if you guys ever want to do this thing, please make sure you discuss it with me....the dates I gave you are always closing down as we get closer so you need to plan ahead.

just fyi,
-paul

EXHIBIT 15

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

SYNOPSYS, INC., a Delaware corporation, Plaintiff and Counter-Defendant, v. MAGMA DESIGN AUTOMATION, a Delaware corporation, Defendant and Counterclaimant.	Case No. 05-701 GMS
AND RELATED COUNTERCLAIMS	

PLAINTIFF SYNOPSYS, INC.'S THIRD SUPPLEMENTAL RESPONSES TO MAGMA DESIGN AUTOMATION, INC.'S FIRST SET OF INTERROGATORIES PURSUANT TO FED. R. CIV. P. 33 (NO. 11)

Dates Entered _____

_____

_____

FEB 26 2007

CALENDARED BY _____

ATTORNEY _____

DECHERT LLP

Pursuant to Rule 33 of the Federal Rules of Civil Procedure, Plaintiff Synopsys, Inc. ("Synopsys") hereby supplements its Responses to Defendant Magma Design Automation, Inc.'s ("Magma") First Set of Interrogatories (No. 11) as follows:

GENERAL OBJECTIONS

1. Synopsys has not fully completed its investigation of the facts relating to this case, has not completed discovery, and has not completed its preparation for trial. Information provided is that which is presently available to and specifically located by Synopsys and its attorneys. Synopsys reserves the right to produce, at the time of trial, additional information which may be discovered subsequent to the filing of these responses.

2. Synopsys objects to each and every interrogatory to the extent that any calls for information to which Magma has equal or greater access than Synopsys.

3. Synopsys objects to each and every interrogatory to the extent that they are vague and ambiguous.

4. Synopsys objects to each and every interrogatory to the extent that any interrogatory calls for information protected by the attorney work product doctrine and/or the attorney-client privilege or any other applicable privilege. To the extent that Synopsys provides any information subject to the attorney-client privilege, the work product doctrine, or other applicable privilege, such disclosure is inadvertent and does not constitute a general waiver of the privilege. Nothing contained herein is intended to be or should be construed as a waiver of the attorney-client privilege or work product doctrine, or any other applicable privilege, protection or doctrine.

5. Synopsys objects to each and every interrogatory to the extent that any interrogatory seeks confidential business information or financial information, trade secrets or

other confidential and protected information.

6. Synopsys objects to each and every interrogatory on the grounds that Magma has yet to produce documents and other information necessary to answer these requests as required by the Federal Rules of Civil Procedure and the Court's local rules.

7. Synopsys objects to each and every interrogatory to the extent the discovery sought is unreasonably cumulative or duplicative, or is obtainable from some other source, including but not limited to Magma itself, that is more convenient, less burdensome, or less expensive. Synopsys also objects to these interrogatories to the extent that the burden or expense of the discovery sought outweighs its likely benefit.

8. Synopsys generally objects to these interrogatories, and to Magma's instructions and definitions, to the extent that any instruction or any definition attempts to impose obligations beyond those imposed under the Federal Rules of Civil Procedure or the Court's local rules.

9. Synopsys objects to the definitions of the terms "MAGMA," "YOU," "YOUR," "SYNOPSYS," "Action," "relate," "relates," and "relating" because they are vague, ambiguous, and compound, and render the Interrogatories that utilize these terms impermissibly vague, ambiguous, and compound in violation of the Federal Rules.

The above general objections are incorporated by reference into each response below.

RESPONSES TO INTERROGATORIES

INTERROGATORY NO. 11:

For each claim in the Magma Patent, state all facts supporting any contention by Synopsys that the claim is invalid or unenforceable.

RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference the General Objections set forth above.

Synopsys objects to this Interrogatory on the grounds that contention-based discovery is premature as discovery in this action has only just started. Synopsys further objects to this Interrogatory on the ground that asking for "all facts" is unduly burdensome. Synopsys further objects to this interrogatory on the ground that the terms "Synopsys Patents," "YOU," and "YOUR" are overbroad, vague and ambiguous. Synopsys objects to this Interrogatory on the grounds that it is compound. Synopsys further objects to this interrogatory to the extent that it calls for information protected from disclosure by the attorney-client privilege and attorney work product doctrine. Synopsys further objects to this request to the extent it seeks confidential, proprietary and/or trade secret information of Synopsys.

FIRST SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys' First Supplemental Responses to Magma's First Set of Interrogatories did not include any supplemental response to Interrogatory No. 11.

SECOND SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference all foregoing specific and General Objections, and provides the following supplemental response subject to and without waiving the foregoing specific and General Objections. Further, the following response is intended to supplement, not supersede, Synopsys' original response. Pursuant to Rule 33(d) of the Federal Rules of Civil Procedure, Synopsys responds as follows:

Synopsys incorporates by reference as though set forth herein its response served on July 7, 2006, and any supplemental responses it may later provide, to Magma Interrogatory No. 13.

THIRD SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference all foregoing specific and General Objections, and provides the following third supplemental response subject to and without waiving the

foregoing specific and General Objections. Further, the following response is intended to supplement, not supersede, Synopsys' prior responses to this interrogatory. Pursuant to Rule 33(d) of the Federal Rules of Civil Procedure, Synopsys responds as follows:

On January 31, 2007, one of the named inventors of the Magma Patent, Lukas van Ginneken, was deposed. During his deposition, Dr. van Ginneken admitted to facts establishing that the Magma Patent (previously defined as U.S. Patent No. 6,505,328) was inequitably procured and, therefore, is unenforceable. As explained below, Dr. van Ginneken admitted that he used technology developed by his previous employer, IBM Corporation ("IBM"), in preparing the patent application that would later issue as the Magma Patent (the "'328 Application"). He also admitted that while this IBM technology is featured in the Magma Patent, he never disclosed its origin to the PTO or to his co-inventors on the Magma Patent. The pre-existing IBM technology used by Dr. van Ginneken in the Magma Patent was prior art to the alleged invention of the '328 Application, was material to the patentability of the claims in the '328 Application, and should have been disclosed to the PTO. Dr. van Ginneken violated his duty of candor to the PTO in failing to disclose this IBM technology to the PTO during prosecution. Moreover, the fact that Dr. van Ginneken knowingly passed off the IBM technology in the Magma Patent as his own inventive work demonstrates an intent to deceive the PTO. Further details of the circumstances giving rise to Dr. van Ginneken's inequitable conduct are set forth below.

In January, 2007, Synopsys noticed for the first time that Figure 4 of the Magma Patent includes some terminology that was also used by IBM in the early 1990's to describe a common data model called the VIDAS In-Core Model ("VIM"). (*See, e.g.,* SYN0018426-429; and SYN0018886-893.) Figure 4 of the Magma Patent reflects the subject matter of the claims because it shows the "data model according to the present invention." (Magma Patent at 3:38-

39.) More specifically, Figure 4 uses the term "definition box" to describe an entity (*see* element 40), the term "proto box" to describe a model (*see* element 50), and the term "usage box" to describe a cell (*see* element 62). (Id., Figure 4.) These terms describe corresponding elements in the VIM common data model that is part of the prior art IBM Technology referred to above. (*See, e.g.,* SYN0018426-429; and SYN0018886-893.)

After first noticing the foregoing in January 2007, Synopsys then reexamined the Magma source code for the product that embodies the Magma Patent and realized that it also includes a number of comments using some of the same IBM terms included in Figure 4. (*See* file named "Bedrock.h," MAGMA0021727A.0001-0026 at 4/26, 7/26 and 14/26). These comments characterize the terms in Figure 4 of the Magma Patent as "VIM terminology." Id. Magma's source code repository suggests that these comments could have been entered by any of several people including Dr. van Ginneken. Dr. van Ginneken has since confirmed that he is the person who entered these comments into the Magma source code.

On January 12, 2007, Synopsys took the deposition of Premal Buch, who had been designated by Magma as a 30(b)(6) witness on a single deposition topic related to the `328 Patent, namely "Magma's use of the Octtools source code in developing the Magma Source Code." (*See* Second Amended Notice of Deposition of Magma Design Automation, Inc., Topic No. 15, p. 9.) When asked who entered the comments using the VIM terminology into the Magma Source code, Dr. Buch testified that he did not know. (Buch Depo., 189:24-196:4.) Until last week, Magma refused to produce a witness on the remaining 30(b)(6) topics, including the topic of "Magma's use of IDM in developing the Magma Source Code." (*See* Second Amended Notice of Deposition of Magma Design Automation, Inc., Topic No. 16, p. 9.) Now that Magma appears to be willing to produce a witness on this topic, Synopsys will take a

deposition on this remaining 30(b)(6) as soon as possible.

The next opportunity to conduct discovery on the Magma Patent came on January 31, 2007, during the deposition of Dr. van Ginneken. Dr. van Ginneken testified that he worked in IBM's logic synthesis group in Yorktown Heights, New York, from 1989 until 1995. (van Ginneken Depo., 6:24-8:3.) During this time, he became very familiar with the VIM data model. (Id., 12:19-14:13.) He testified that he gained first hand experience in using VIM as a common data model that was shared by a number of IBM design tools, including: a logic synthesis tool called Booledozer; a suite of physical design tools called VIDAS; and at least two different static timing analysis tools called Einstimer and Slackhoe. Dr. van Ginneken admitted that, like the alleged invention of the Magma Patent, IBM's prior art VIM data model is a "common data model." (Id.)

In 1994, Dr. van Ginneken and Robert Damiano published a paper entitled "Timing Verification and Optimization For the PowerPC Processor Family," IEEE © 1994 (the "PowerPC Paper"). (van Ginneken Depo., 31:9-24.) This paper focuses on the use of Booledozer and Einstimer in developing a family of microprocessors. (Id., 32:21-37:18.) The paper also suggests that these tools shared a single data model, but it does not disclose the name of the single data model or any details. (Id.) However, Dr. van Ginneken admitted during his deposition that the data model mentioned in this publication was the VIM common data model. (Id.)

During his employment at IBM, Dr. van Ginneken also became aware that IBM was in the process of developing another, more advanced common data model called the Integrated Data Model ("IDM"). (van Ginneken Depo., 11:3-12.) He testified that IDM was developed by a group at IBM called the electronic design automation ("EDA") Group, which was based in

Fishkill, New York. (Id..) Dr. van Ginneken said he was aware that IDM was derived from VIM. (Id., 14:5-23.)

Dr. van Ginneken also admitted that he knew about an initiative, endorsed by IBM, to establish an industry standard common data model based on IBM's VIM common data model. (van Ginneken Depo., 16:5-17:8.) This standard was promoted by two EDA industry consortiums, Silicon Integration Initiative ("SI2") and SEMATECH, working in partnership during the mid to late 1990's. This initiative included the promotion of a chip design system called the Chip Hierarchical Design System ("CHDS"). The goal of CHDS was to provide flexible design methodologies for timing-driven logical and physical design of complex ICs. As part of the CHDS development, SEMATECH and SI2 also developed a high-performance hierarchical design representation standard, called the CHDS Technical Data ("CHDStd"). Dr. van Ginneken testified that he knew IBM donated IDM (a derivative of VIM) to be used as a basis for the development of CHDStd. (Id., 116:5-13.) Dr. van Ginneken further admitted that knew about the effort to establish IDM as an industry-wide standard common data model before he filed the application for the Magma Patent. (Id., 116:21-117:1.)

Dr. van Ginneken testified that he began working on a unified data model immediately after co-founding Magma in the spring of 1997. (van Ginneken Depo., 81:24-82:23.) He further testified that he was the principal author of the Magma source code file named "Bedrock.h," which implements Magma's unified data model. (Id., 53:8-54:1.) When confronted with the comments in the Magma source code referring to the VIM terminology, Dr. van Ginneken admitted that he was the one who entered these comments. (Id., 53:8-59:16.) He testified that no one else at Magma, other than him, would have understood these comments. (Id., 58:7-23.) Dr. van Ginneken said his purpose in entering these comments was to clarify the source code for

himself. (Id., 56:4-20.) Therefore, he was not only cognizant of IBM's common data model while developing Magma's unified data model, he actually found it useful to document the source code to reflect the correlations between the two common data models. (Id.)

Dr. van Ginneken also admitted that the terminology shown in Figure 4 of the Magma Patent (*i.e.*, "proto box" "usage box" etc.) is indeed the same VIM terminology that he learned about while he worked at IBM. (van Ginneken Depo., 64:1-65:8.) He testified that the text in Figure 4 must have been provided to patent prosecution counsel by him because none of his other colleagues at Magma were familiar with VIM. (Id., 132:18-133:7.) And despite his knowledge of the prior art VIM common data model, Dr. van Ginneken endorsed the statements in the background section of the `328 Application, which suggest that in the prior art, "a separate internal data structure is used for each tool." (Magma Patent at 2:23-25.) Therefore, not only was Dr. van Ginneken aware of a prior art common data model that he never disclosed to his colleagues, his patent attorney or the PTO, he purposefully tried to deceive his colleagues, his patent attorney and the PTO into believing that the closest prior art used a separate internal data structure for each tool. As explained above, Dr. van Ginneken knew full well that the IBM common data model was far more relevant to the claimed invention than the prior art discussed in the background section of the Magma Patent.

In early 2003, Dr. van Ginneken co-authored a paper entitled "Design Flow and Methodology for 50M gate ASIC" © 2003 (the "Design Flow Paper"). This paper describes a technology that leverages "a common unified data model," which refers to Magma's own unified data model, which is the subject of the Magma Patent. (*See* SYN1616527-534). The "Previous Work" section of this paper cites to another paper entitled "Chip Hierarchical Design System (CHDS): A foundation for Timing-Driven Physical Design into the 21st Century," Bushroe et al.,

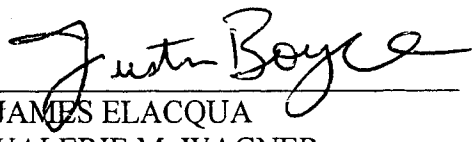
Proc. Int. Symp. Physical Design, pp. 212-217, 1997 (the "CHDS Foundation Paper"). (Id. at p. 640). The CHDS Foundation Paper describes the use of IDM as a common data model, and also cites to numerous publications that were available before the `328 Application was filed. These cited publications describe the technical details of both IDM and VIM, including the use of a tree data structure to perform area query operations on IDM, which was later claimed in the Magma Patent as the point of novelty. (*See* SI20001-Si20006.) While the Design Flow Paper was published shortly after the Magma Patent issued on January 7, 2003, the deposition transcript of Dr. van Ginneken reveals that he personally completed the Previous Work section of the Design Flow Paper on November 8, 2002, before the Magma Patent issued. (van Ginneken Depo., 111:16-114:10.) Dr. van Ginneken's authorship of the Design Flow Paper, when viewed in light of his knowledge of IDM and VIM, clearly indicates that during the pendency of the `328 Application: (1) Dr. van Ginneken considered the prior art IDM and VIM common data models to be relevant to his own work on Magma's common data model; (2) Dr. van Ginneken knew where materials describing the prior art VIM and IDM common data models were located, and (3) Dr. van Ginneken knew that area queries could be made on the prior art IDM common data model as claimed in the Magma Patent. In light of the foregoing, Dr. van Ginneken's failure to disclose any information about the prior art VIM and IDM common data models to the PTO must have been deliberate and undertaken with an intent to deceive the PTO into granting the Magma Patent.

Dr. van Ginneken's deceptive misconduct as set forth above renders the Magma Patent unenforceable. But there is further circumstantial evidence of Dr. van Ginneken's intent to act deceitfully to secure patents on behalf of Magma based on work performed by others at a previous place of employment. More specifically, Dr. van Ginneken previously admitted to

using technology developed while he was employed by Synopsys in a Magma patent application that he later filed in his own name, and assigned to Magma. (van Ginneken Depo., 49:16-52:12.) Indeed, Dr. van Ginneken signed a sworn declaration describing the facts and circumstances surrounding his illicit use of technology owned by Synopsys in two different U.S. Patents he assigned to Magma; namely U.S. Patent No. 6,453,446 (the "`446 Patent") and U.S. Patent No. 6,725,438 (the "`438 Patent"). (Id., 152:2-153:3, discussing Synopsys' Deposition Exhibit 396.) Dr. van Ginneken's passing off of IBM's technology as an invention he purportedly conceived while working for Magma fits the same pattern and practice demonstrated by his actions in connection with the `446 and `438 Patents.

DATED: February 26, 2007

DECHERT LLP

By 

JAMES ELACQUA
VALERIE M. WAGNER
JUSTIN F. BOYCE
Attorneys for Plaintiff and Cross-Defendant SYNOPSYS, INC.

Jack B. Blumenfield (#1014)
Karen Jacobs Louden (#2881)
Klouden@mnat.com
MORRIS, NICHOLS, ARSHT & TUNNELL
1201 N. Market Street
P.O. Box 1347
Wilmington, DE 19899-1347
(302) 658-9200

CERTIFICATE OF SERVICE

I HEREBY CERTIFY that on February 26, 2007, I served the foregoing document on counsel of record in the manner indicated.

VIA ELECTRONIC MAIL AND HAND DELIVERY

Pooley & Oliver, LLP
Five Palo Alto Square, 7th Floor
3000 El Camino Real
Palo Alto, CA 94306-2109
(courtesy copy by facsimile)

VIA ELECTRONIC MAIL AND U.S. MAIL

William J. Marsden, Jr.
Fish & Richardson P.C.
919 N. Market Street
P.O. Box 1114
Wilmington, DE 19899

Justin F. Boyce

DATED: February 26, 2007

EXHIBIT 16

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

SYNOPSYS, INC., a Delaware corporation, Plaintiff and Counter-Defendant, v. MAGMA DESIGN AUTOMATION, a Delaware corporation, Defendant and Counterclaimant. AND RELATED COUNTERCLAIMS	Case No. 05-701 GMS

Dates Entered please Advise

MAR 0 5 2007

CALENDARED BY jmh

ATTORNEY

DECHERT LLP

PLAINTIFF SYNOPSYS, INC.'S FOURTH SUPPLEMENTAL RESPONSES TO MAGMA DESIGN AUTOMATION, INC.'S FIRST SET OF INTERROGATORIES PURSUANT TO FED. R. CIV. P. 33 (NO. 11)

Pursuant to Rule 33 of the Federal Rules of Civil Procedure, Plaintiff Synopsys, Inc. ("Synopsys") hereby supplements its Responses to Defendant Magma Design Automation, Inc.'s ("Magma") First Set of Interrogatories (No. 11) as follows:

GENERAL OBJECTIONS

1. Synopsys has not fully completed its investigation of the facts relating to this case, has not completed discovery, and has not completed its preparation for trial. Information provided is that which is presently available to and specifically located by Synopsys and its attorneys. Synopsys reserves the right to produce, at the time of trial, additional information which may be discovered subsequent to the filing of these responses.

2. Synopsys objects to each and every interrogatory to the extent that any calls for information to which Magma has equal or greater access than Synopsys.

3. Synopsys objects to each and every interrogatory to the extent that they are vague and ambiguous.

4. Synopsys objects to each and every interrogatory to the extent that any interrogatory calls for information protected by the attorney work product doctrine and/or the attorney-client privilege or any other applicable privilege. To the extent that Synopsys provides any information subject to the attorney-client privilege, the work product doctrine, or other applicable privilege, such disclosure is inadvertent and does not constitute a general waiver of the privilege. Nothing contained herein is intended to be or should be construed as a waiver of the attorney-client privilege or work product doctrine, or any other applicable privilege, protection or doctrine.

5. Synopsys objects to each and every interrogatory to the extent that any interrogatory seeks confidential business information or financial information, trade secrets or other confidential and protected information.

6. Synopsys objects to each and every interrogatory on the grounds that Magma has yet to produce documents and other information necessary to answer these requests as required by the Federal Rules of Civil Procedure and the Court's local rules.

7. Synopsys objects to each and every interrogatory to the extent the discovery sought is unreasonably cumulative or duplicative, or is obtainable from some other source, including but not limited to Magma itself, that is more convenient, less burdensome, or less expensive. Synopsys also objects to these interrogatories to the extent that the burden or expense of the discovery sought outweighs its likely benefit.

8. Synopsys generally objects to these interrogatories, and to Magma's instructions and definitions, to the extent that any instruction or any definition attempts to impose obligations beyond those imposed under the Federal Rules of Civil Procedure or the Court's local rules.

9. Synopsys objects to the definitions of the terms "MAGMA," "YOU," "YOUR," "SYNOPSYS," "Action," "relate," "relates," and "relating" because they are vague, ambiguous, and compound, and render the Interrogatories that utilize these terms impermissibly vague, ambiguous, and compound in violation of the Federal Rules.

The above general objections are incorporated by reference into each response below.

RESPONSES TO INTERROGATORIES

INTERROGATORY NO. 11:

For each claim in the Magma Patent, state all facts supporting any contention by Synopsys that the claim is invalid or unenforceable.

RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference the General Objections set forth above. Synopsys objects to this Interrogatory on the grounds that contention-based discovery is premature as discovery in this action has only just started. Synopsys further objects to this

Interrogatory on the ground that asking for "all facts" is unduly burdensome. Synopsys further objects to this interrogatory on the ground that the terms "Synopsys Patents," "YOU," and "YOUR" are overbroad, vague and ambiguous. Synopsys objects to this Interrogatory on the grounds that it is compound. Synopsys further objects to this interrogatory to the extent that it calls for information protected from disclosure by the attorney-client privilege and attorney work product doctrine. Synopsys further objects to this request to the extent it seeks confidential, proprietary and/or trade secret information of Synopsys.

FIRST SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys' First Supplemental Responses to Magma's First Set of Interrogatories did not include any supplemental response to Interrogatory No. 11.

SECOND SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference all foregoing specific and General Objections, and provides the following supplemental response subject to and without waiving the foregoing specific and General Objections. Further, the following response is intended to supplement, not supersede, Synopsys' original response. Pursuant to Rule 33(d) of the Federal Rules of Civil Procedure, Synopsys responds as follows:

Synopsys incorporates by reference as though set forth herein its response served on July 7, 2006, and any supplemental responses it may later provide, to Magma Interrogatory No. 13.

THIRD SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference all foregoing specific and General Objections, and provides the following third supplemental response subject to and without waiving the foregoing specific and General Objections. Further, the following response is intended to supplement, not supersede, Synopsys' prior responses to this interrogatory. Pursuant to Rule 33(d) of the Federal Rules of Civil Procedure, Synopsys responds as follows:

On January 31, 2007, one of the named inventors of the Magma Patent, Lukas van Ginneken, was deposed. During his deposition, Dr. van Ginneken admitted to facts establishing that the Magma Patent (previously defined as U.S. Patent No. 6,505,328) was inequitably procured and, therefore, is unenforceable. As explained below, Dr. van Ginneken admitted that he used technology developed by his previous employer, IBM Corporation ("IBM"), in preparing the patent application that would later issue as the Magma Patent (the "'328 Application"). He also admitted that while this IBM technology is featured in the Magma Patent, he never disclosed its origin to the PTO or to his co-inventors on the Magma Patent. The pre-existing IBM technology used by Dr. van Ginneken in the Magma Patent was prior art to the alleged invention of the '328 Application, was material to the patentability of the claims in the '328 Application, and should have been disclosed to the PTO. Dr. van Ginneken violated his duty of candor to the PTO in failing to disclose this IBM technology to the PTO during prosecution. Moreover, the fact that Dr. van Ginneken knowingly passed off the IBM technology in the Magma Patent as his own inventive work demonstrates an intent to deceive the PTO. Further details of the circumstances giving rise to Dr. van Ginneken's inequitable conduct are set forth below.

In January, 2007, Synopsys noticed for the first time that Figure 4 of the Magma Patent includes some terminology that was also used by IBM in the early 1990's to describe a common data model called the VIDAS In-Core Model ("VIM"). (*See, e.g.*, SYN0018426-429; and SYN0018886-893.) Figure 4 of the Magma Patent reflects the subject matter of the claims because it shows the "data model according to the present invention." (Magma Patent at 3:38-39.) More specifically, Figure 4 uses the term "definition box" to describe an entity (*see* element 40), the term "proto box" to describe a model (*see* element 50), and the term "usage box" to describe a cell (*see* element 62). (Id., Figure 4.) These terms describe corresponding

elements in the VIM common data model that is part of the prior art IBM Technology referred to above. (*See, e.g.*, SYN0018426-429; and SYN0018886-893.)

After first noticing the foregoing in January 2007, Synopsys then reexamined the Magma source code for the product that embodies the Magma Patent and realized that it also includes a number of comments using some of the same IBM terms included in Figure 4. (*See* file named "Bedrock.h," MAGMA0021727A.0001-0026 at 4/26, 7/26 and 14/26). These comments characterize the terms in Figure 4 of the Magma Patent as "VIM terminology." Id. Magma's source code repository suggests that these comments could have been entered by any of several people including Dr. van Ginneken. Dr. van Ginneken has since confirmed that he is the person who entered these comments into the Magma source code.

On January 12, 2007, Synopsys took the deposition of Premal Buch, who had been designated by Magma as a 30(b)(6) witness on a single deposition topic related to the '328 Patent, namely "Magma's use of the Octtools source code in developing the Magma Source Code." (*See* Second Amended Notice of Deposition of Magma Design Automation, Inc., Topic No. 15, p. 9.) When asked who entered the comments using the VIM terminology into the Magma Source code, Dr. Buch testified that he did not know. (Buch Depo., 189:24-196:4.) Until last week, Magma refused to produce a witness on the remaining 30(b)(6) topics, including the topic of "Magma's use of IDM in developing the Magma Source Code." (*See* Second Amended Notice of Deposition of Magma Design Automation, Inc., Topic No. 16, p. 9.) Now that Magma appears to be willing to produce a witness on this topic, Synopsys will take a deposition on this remaining 30(b)(6) as soon as possible.

The next opportunity to conduct discovery on the Magma Patent came on January 31, 2007, during the deposition of Dr. van Ginneken. Dr. van Ginneken testified that he worked in IBM's logic synthesis group in Yorktown Heights, New York, from 1989 until 1995. (van

Ginneken Depo., 6:24-8:3.) During this time, he became very familiar with the VIM data model. (Id., 12:19-14:13.) He testified that he gained first hand experience in using VIM as a common data model that was shared by a number of IBM design tools, including: a logic synthesis tool called Booledozer; a suite of physical design tools called VIDAS; and at least two different static timing analysis tools called Einstimer and Slackhoe. Dr. van Ginneken admitted that, like the alleged invention of the Magma Patent, IBM's prior art VIM data model is a "common data model." (Id.)

In 1994, Dr. van Ginneken and Robert Damiano published a paper entitled "Timing Verification and Optimization For the PowerPC Processor Family," IEEE © 1994 (the "PowerPC Paper"). (van Ginneken Depo., 31:9-24.) This paper focuses on the use of Booledozer and Einstimer in developing a family of microprocessors. (Id., 32:21-37:18.) The paper also suggests that these tools shared a single data model, but it does not disclose the name of the single data model or any details. (Id.) However, Dr. van Ginneken admitted during his deposition that the data model mentioned in this publication was the VIM common data model. (Id.)

During his employment at IBM, Dr. van Ginneken also became aware that IBM was in the process of developing another, more advanced common data model called the Integrated Data Model ("IDM"). (van Ginneken Depo., 11:3-12.) He testified that IDM was developed by a group at IBM called the electronic design automation ("EDA") Group, which was based in Fishkill, New York. (Id.) Dr. van Ginneken said he was aware that IDM was derived from VIM. (Id., 14:5-23.)

Dr. van Ginneken also admitted that he knew about an initiative, endorsed by IBM, to establish an industry standard common data model based on IBM's VIM common data model. (van Ginneken Depo., 16:5-17:8.) This standard was promoted by two EDA industry

consortiums, Silicon Integration Initiative ("SI2") and SEMATECH, working in partnership during the mid to late 1990's. This initiative included the promotion of a chip design system called the Chip Hierarchical Design System ("CHDS"). The goal of CHDS was to provide flexible design methodologies for timing-driven logical and physical design of complex ICs. As part of the CHDS development, SEMATECH and SI2 also developed a high-performance hierarchical design representation standard, called the CHDS Technical Data ("CHDStd"). Dr. van Ginneken testified that he knew IBM donated IDM (a derivative of VIM) to be used as a basis for the development of CHDStd. (Id., 116:5-13.) Dr. van Ginneken further admitted that knew about the effort to establish IDM as an industry-wide standard common data model before he filed the application for the Magma Patent. (Id., 116:21-117:1.)

Dr. van Ginneken testified that he began working on a unified data model immediately after co-founding Magma in the spring of 1997. (van Ginneken Depo., 81:24-82:23.) He further testified that he was the principal author of the Magma source code file named "Bedrock.h," which implements Magma's unified data model. (Id., 53:8-54:1.) When confronted with the comments in the Magma source code referring to the VIM terminology, Dr. van Ginneken admitted that he was the one who entered these comments. (Id., 53:8-59:16.) He testified that no one else at Magma, other than him, would have understood these comments. (Id., 58:7-23.) Dr. van Ginneken said his purpose in entering these comments was to clarify the source code for himself. (Id., 56:4-20.) Therefore, he was not only cognizant of IBM's common data model while developing Magma's unified data model, he actually found it useful to document the source code to reflect the correlations between the two common data models. (Id.)

Dr. van Ginneken also admitted that the terminology shown in Figure 4 of the Magma Patent (*i.e.*, "proto box" "usage box" etc.) is indeed the same VIM terminology that he learned about while he worked at IBM. (van Ginneken Depo., 64:1-65:8.) He testified that the text in

Figure 4 must have been provided to patent prosecution counsel by him because none of his other colleagues at Magma were familiar with VIM. (Id., 132:18-133:7.) And despite his knowledge of the prior art VIM common data model, Dr. van Ginneken endorsed the statements in the background section of the '328 Application, which suggest that in the prior art, "a separate internal data structure is used for each tool." (Magma Patent at 2:23-25.) Therefore, not only was Dr. van Ginneken aware of a prior art common data model that he never disclosed to his colleagues, his patent attorney or the PTO, he purposefully tried to deceive his colleagues, his patent attorney and the PTO into believing that the closest prior art used a separate internal data structure for each tool. As explained above, Dr. van Ginneken knew full well that the IBM common data model was far more relevant to the claimed invention than the prior art discussed in the background section of the Magma Patent.

In early 2003, Dr. van Ginneken co-authored a paper entitled "Design Flow and Methodology for 50M gate ASIC" © 2003 (the "Design Flow Paper"). This paper describes a technology that leverages "a common unified data model," which refers to Magma's own unified data model, which is the subject of the Magma Patent. (*See* SYN1616527-534). The "Previous Work" section of this paper cites to another paper entitled "Chip Hierarchical Design System (CHDS): A foundation for Timing-Driven Physical Design into the 21st Century," Bushroe et al., Proc. Int. Symp. Physical Design, pp. 212-217, 1997 (the "CHDS Foundation Paper"). (Id. at p. 640). The CHDS Foundation Paper describes the use of IDM as a common data model, and also cites to numerous publications that were available before the '328 Application was filed. These cited publications describe the technical details of both IDM and VIM, including the use of a tree data structure to perform area query operations on IDM, which was later claimed in the Magma Patent as the point of novelty. (*See* SI20001-Si20006.) While the Design Flow Paper was published shortly after the Magma Patent issued on January 7, 2003, the deposition transcript of

Dr. van Ginneken reveals that he personally completed the Previous Work section of the Design Flow Paper on November 8, 2002, before the Magma Patent issued. (van Ginneken Depo., 111:16-114:10.) Dr. van Ginneken's authorship of the Design Flow Paper, when viewed in light of his knowledge of IDM and VIM, clearly indicates that during the pendency of the '328 Application: (1) Dr. van Ginneken considered the prior art IDM and VIM common data models to be relevant to his own work on Magma's common data model; (2) Dr. van Ginneken knew where materials describing the prior art VIM and IDM common data models were located, and (3) Dr. van Ginneken knew that area queries could be made on the prior art IDM common data model as claimed in the Magma Patent. In light of the foregoing, Dr. van Ginneken's failure to disclose any information about the prior art VIM and IDM common data models to the PTO must have been deliberate and undertaken with an intent to deceive the PTO into granting the Magma Patent.

Dr. van Ginneken's deceptive misconduct as set forth above renders the Magma Patent unenforceable. But there is further circumstantial evidence of Dr. van Ginneken's intent to act deceitfully to secure patents on behalf of Magma based on work performed by others at a previous place of employment. More specifically, Dr. van Ginneken previously admitted to using technology developed while he was employed by Synopsys in a Magma patent application that he later filed in his own name, and assigned to Magma. (van Ginneken Depo., 49:16-52:12.) Indeed, Dr. van Ginneken signed a sworn declaration describing the facts and circumstances surrounding his illicit use of technology owned by Synopsys in two different U.S. Patents he assigned to Magma; namely U.S. Patent No. 6,453,446 (the "'446 Patent") and U.S. Patent No. 6,725,438 (the "'438 Patent"). (Id., 152:2-153:3, discussing Synopsys' Deposition Exhibit 396.) Dr. van Ginneken's passing off of IBM's technology as an invention he purportedly conceived

while working for Magma fits the same pattern and practice demonstrated by his actions in connection with the '446 and '438 Patents.

FOURTH SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 11:

Synopsys hereby incorporates by reference all foregoing specific and General Objections and provides the following fourth supplemental response subject to and without waiving the foregoing specific and General Objections. Furthermore, the following response is intended to supplement, not supersede, Synopsys's prior responses to this interrogatory. Pursuant to Rule 33(b) of the Federal Rules of Civil Procedure, Synopsys responds as follows:

The subject matter of at least method claims 1-10 (and, by implication, at least the corresponding "computer-readable medium" claims 11- 20) of the '093 patent, as well as the subject matter of at least claims 2-21 of U.S. Application 10/104,813 submitted on March 22, 2002, was embodied in ReShape's software before the November 15, 1999 critical date of the '093 patent. (*See, e.g.*, '093 Patent; Synopsys Exs. 328, 329, 398; Levine Dep. 160:22-166:7; 209:19-210:12, 243:7-245:22, 248:20-249:13, 250:5-17; Dickinson Dep. (unofficial transcript) 153:21-157:17, 168:2-180:7; *see also* Dahl Dep.) The subject matter of at least claims 2-21 of U.S. Application 10/104,813 submitted on March 22, 2002 and of at least claims 1-20 of the '093 patent was also embodied in ReShape's services performed before that critical date. (*See, e.g.*, '093 Patent; Synopsys Ex. 314; Levine Dep. 84:17-85:18, 111:15-112:23, 113:12-114:2; Dahl Dep.)

The subject matter of at least method claims 1, 2, and 7-14 (and, by implication, at least corresponding "computer-readable medium" claims 15, 16, and 21-28) of the '116 patent, as well as the subject matter of originally filed claim 1 and at least claims 2, 5, 6, 9-16, 19, 20, and 23-28 of U.S. Application 09/714,722 submitted on March 22, 2002, was embodied in ReShape's software before the November 15, 1999 critical date of the '116 patent. (*See, e.g.*, '116 Patent;

Synopsys Exs. 315, 323, 328, 329, 398-406; Levine Dep. 97:8-99:11, 127:22-131:11, 160:22-166:7, 209:19-210:12, 254:10-256:3; 227:12-230:2; 234:7-239:6, 243:7-245:22, 248:20-249:13; Dickinson Dep. (unofficial transcript) 153:21-157:17, 168:2-174:16; Dahl Dep.) The subject matter of originally filed claim 1, at least claims 2, 5, 6, 9-16, 19, 20, and 23-28 of U.S. Application 09/714,722 submitted on March 22, 2002, and of at least claims 1, 2, 7-16, and 21-28 of the '116 patent was also embodied in ReShape's services performed before the November 15, 1999 critical date. (*See, e.g.*, '116 Patent; Synopsys Ex. 314; Levine Dep. 84:17-85:18, 111:15-112:23, 113:12-114:5, 167:21-174:7, 176:24-177:14, 227:12-230:1; 234:7-239:6; Dickinson Dep. (unofficial transcript) 153:21-157:17, 168:2-174:16; *see also* Dahl Dep.)

The above described ReShape software and services were sold and offered for sale by one or more of the individuals named as inventors on the '093 and '116 patents before the November 15, 1999 critical date for both the '093 and '116 patents. (*See, e.g.*, Synopsys Exs. 314, 315, 323; Levine Dep. 84:17-85:18, 97:8-99:11, 111:15-112:23, 113:12-114:5, 127:22-131:11, 167:21-173:1, 254:10-256:3.)

The evidence thus clearly and convincingly shows that the subject matter of at least one claim of the '093 patent and at least one claim of the '116 patent was both offered for sale and sold before the critical date of the patent, rendering those claims invalid under 35 U.S.C. § 102(b). The evidence also clearly and convincingly shows that individuals named as inventors on the '093 and '116 patents were aware of that invalidating sales activity. The evidence further shows the applicants (and other individuals who were substantively involved with the preparation or prosecution of the applications as defined by 37 CFR § 1.56) intentionally failed to satisfy the duty of candor and full and complete disclosure to the U.S. Patent and Trademark Office (USPTO) during the prosecution of the applications that issued as the '093 and '116 patents by not disclosing that invalidating sales activity to the USPTO. In so doing, the

applicants violated their April 16, 2001 oaths filed in U.S. Applications 09/714,722 and 10/104,813. Because that sales activity rendered unpatentable the subject matter for which the patents were sought, ReShape's sale and offer for sale were of the highest possible materiality to the prosecution of the applications that issued as the '093 and '116 patents, and ReShape's failure to disclose any of that activity to the PTO raises an inference of intent to deceive the PTO. Accordingly, the intentional failure to disclose any of that material information was inequitable conduct, making the '093 and '116 patents unenforceable in their entirety.

DATED: March 2, 2007

DECHERT LLP

By _____

JAMES J. ELACQUA
VALERIE M. WAGNER
CONNIE E. MERRIETT
Attorneys for Plaintiff and Cross-Defendant SYNOPSYS, INC.

Jack B. Blumenfield (#1014)
Karen Jacobs Louden (#2881)
Klouden@mnat.com
MORRIS, NICHOLS, ARSHT & TUNNELL
1201 N. Market Street
P.O. Box 1347
Wilmington, DE 19899-1347
(302) 658-9200

CERTIFICATE OF SERVICE

I HEREBY CERTIFY that on March 2, 2007, I served copies of PLAINTIFF SYNOPSYS, INC.'S FOURTH SUPPLEMENTAL RESPONSES TO MAGMA DESIGN AUTOMATION, INC.'S FIRST SET OF INTERROGATORIES PURSUANT TO FED. R. CIV. P. 33 (NO. 11) on counsel of record in the manner indicated.

VIA ELECTRONIC MAIL AND U.S. MAIL

L. Scott Oliver, Esq.
Pooley & Oliver, LLP
Five Palo Alto Square, 7th Floor
3000 El Camino Real
Palo Alto, CA 94306-2109

William Marsden, Esq.
Fish & Richardson P.C.
919 N. Market Street
P.O. Box 1114
Wilmington, DE 19899

Connie E. Merriett

DATED: March 2, 2007

EXHIBIT 17

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

Dates Entered _____

(previously rec'd by email)

MAR 0 2 2007

CALENDARED BY _____

ATTORNEY _____

DECHERT LLP

SYNOPSYS, INC.,
a Delaware corporation,

Plaintiff and
Counter-Defendant,

v.

MAGMA DESIGN AUTOMATION, INC.,
a Delaware corporation,

Defendant and
Counter-Claimant.

C.A. No. 05-701-GMS

DEFENDANT MAGMA DESIGN AUTOMATION, INC.'S THIRD SUPPLEMENTAL INITIAL DISCLOSURE PURSUANT TO FED. R. CIV. P. 26(a)(1)

Pursuant to Rule 26 of the Federal Rules of Civil Procedure, Defendant and Counterclaimant Magma Design Automation, Inc. ("Magma") hereby provides the following Third Supplemental Initial Disclosure to Synopsys, Inc. ("Synopsys"). This Third Supplemental Initial Disclosure is based on information currently available to Magma and upon a reasonable investigation conducted in the time available. Magma has not yet completed formal discovery. Magma's investigation and analysis is continuing, and Magma reserves the right to supplement, amend, or otherwise modify this Third Supplemental Initial Disclosure based upon further investigation, review, analysis, and discovery. Magma reserves the right to amend this Third Supplemental Initial Disclosure and provide additional information, or to otherwise make additional information known to Synopsys, at appropriate times. By providing this Third Supplemental Initial Disclosure, Magma does not waive and expressly preserves, any and all objections, claims of privilege, and/or work product protections.

I. WITNESSES

Pursuant to Fed. R. Civ. P. 26(a)(1)(A), Magma identifies the following as individuals who may have discoverable information that Magma may use in support of its claims and defenses, excluding those individuals who may have discoverable information that Magma may use solely for impeachment. The indication of knowledge is not intended to be all-inclusive and listed witnesses may have relevant information in other areas as well. Magma reserves the right to identify additional witnesses once formal discovery has been commenced and completed, and at such time that any additional relevant information may become available. Magma also reserves the right to later name expert witnesses for technical and damages issues.

1. Patrick R. Groeneveld

2. Wilhelmus J. M. Philipsen

The subject matter for the foregoing individuals' information is: the conception, reduction to practice, and ownership of U.S. Patent No. 6,505,328, Magma's technology and products, and the field and technology of electronic design automation software. These individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

3. Lukas P. P. P. van Ginneken

The subject matter for the foregoing individual's information is: the conception, reduction to practice, and ownership of U.S. Patent No. 6,505,328, Magma's technology and products, and the field and technology of electronic design automation software. Dr. van Ginneken can be contacted through his counsel, Edward W. Bulchis, Dorsey & Whitney, LLP, U.S. Bank Centre, 1420 Fifth Avenue, Suite 3400, Seattle, WA 98101, (206) 903-8785.

4. Premal Buch

5. Manjit Borah

The subject matter for the foregoing individual's information is: the conception, reduction to practice, and ownership of U.S. Patent Nos. 6,931,610 and 6,519,745, Magma's technology and products, and the field and technology of electronic design automation software. These individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

6. Prasanna Venkat Srinivas

The subject matter for the foregoing individual's information is: the conception, reduction to practice, and ownership of U.S. Patent No. 6,519,745, Magma's technology and products, and the field and technology of electronic design automation software. Dr. Srinivas can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

7. Peter Dahl

8. Byron Dickinson

9. Margie Levine

10. Paul Rodman

The subject matter for the foregoing individuals' information is: the conception, reduction to practice, and ownership of U.S. Patent Nos. 6,857,116 and 6,854,093, and the field and technology of electronic design automation software. The individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

11. Hong Cai

12. Eduard P. Huijbregts

13. Eric Seelen

14. Robert Swanson

15. Samit Chaudhuri

16. Jeffrey Gee

17. Amir Voskoboynik

18. Henrik Esbensen

19. Indresh Arora

20. Rohit Kumar
21. Raghunandan Rao
22. Hardy Kwok-Shing Leung
23. Cornelius A. J. van Eijk
24. Olivier Coudert
25. Vivek Raghavan
26. Dwayne Burek
27. Robert Thompson
28. R. Dean Adams
29. Moez Cherif
30. Djavad Amiri
31. Donnacha O'Riordan
32. Yatin Trivedi
33. Arvind Srinivasan
34. Kam Kittrell
35. Hyukyong Kwon
36. Myoungkoo Jin
37. Eun Sei Park
38. Hsaio-Ping Tseng
39. Leo van Borkhoven
40. Menno Verbeek
41. Tsong Wen Her
42. Tao Lin
43. Srihari Pamanathan
44. Roger Carpenter
45. Sandeep Grover

The subject matter for the foregoing individuals' information is: Magma's technology and products, and the field and technology of electronic design automation software. Mr. Coudert can be contacted through his counsel, James Quadra, Moscone, Emblidge & Quadra, LLP, 180 Montgomery Street, Suite 1240, San Francisco, CA 94104, (415) 362 0265. Mr. Raghavan can be contacted through his counsel, James McManis, McManis, Faulker & Morgan, 50 West San Fernando Street, Suite 1000, 10th Floor, San Jose, CA 95113, (408) 279 8700. The remaining individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

46. Hunaid Hussain

47. Xiaoliang "Shawn" Bai

48. Eric MacDonald

49. Raymond Nijssen

50. Robert Doig

51. Robert Abbott

52. Chidambara Ramaswamy

53. Sumit Kharabanda

54. Hyoung Bok Min

55. Lionel Corbet

56. Felix Huang

57. Stephen McInerney

58. Wein-Jieh Sun

59. Ajit Sequeira

60. PV Srinivas

The subject matter for the foregoing individuals' information is: Magma's technology and products, and the field and technology of electronic design automation software. Magma is not aware of the present addresses or telephone numbers of these individuals.

61. David A. Jakopin

The subject matter for the foregoing individual's information is: prosecution of U.S. Patent Nos. 6,505,328, 6,519,745, and 6,931,610 Mr. Jakopin can be reached at Pillsbury Winthrop LLP, 2475 Hanover Street, Palo Alto, CA 94304, (650) 233-4500.

62. Jose S. Garcia

The subject matter for the foregoing individual's information is: prosecution of U.S. Patent Nos. 6,857,116 and 6,854,093. Mr. Garcia can be reached at Wagner, Murabito & Hao LLP, Two North Market Street, Third Floor, San Jose, CA 95113.

63. Anthony C. Murabito

The subject matter for the foregoing individual's information is: prosecution of U.S. Patent Nos. 6,434,733, 6,766,501, 6,405,355, 6,857,116 and 6,854,093. Mr. Murabito can be reached at Wagner, Murabito & Hao LLP, Two North Market Street, Third Floor, San Jose, CA 95113, (408) 938 9060.

64. Mehlin Dean Matthews

65. Jeanette S. Harms

The subject matter for the foregoing individuals' information is: prosecution of U.S. Patent Nos. 6,434,733, 6,766,501, and 6,405,355. Ms. Harms can be reached at Bever, Hoffman & Harms, LLP, 2099 Gateway Place, Suite 320, San Jose, CA 95110, (408) 451 5907. Mr. Matthews can be reached at Polaris Research, P. O. Box 24, Saratoga, CA 95071, (408) 364 2170.

66. Michael J. Ure

The subject matter for the foregoing individual's information is: the prosecution of U.S. Patent No. 6,192,508. Mr. Ure can be reached at Burns, Doane, Swecker & Mathis, LLP, P.O. Box 104, Alexandria, Virginia 22313-1404.

67. Saeid Ghafouri

68. Roy E. Jewell

69. Gregory C. Walker

70. Hamid Savoj

The subject matter for the foregoing individuals' information is: Synopsys's unfair competition, Synopsys's false and disparaging comments to Magma's customers and the general public, and interference with Magma's business relationships. These individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

71. Christian Landrault, LIRMM

72. Paolo Prinetto, Politecnio di Torino

73. Hans J. Wunderlich, University of Siegen, Germany

74. Kees Baker, Philips

75. Serge Pravossoudovitch, LIRMM

76. Yves Bertrand, LIRMM

77. Michael Nicolaidis, TIMA/INPG

78. Bruno Rouzeyre, LIRMM

The subject matter for the foregoing individuals' information is: proposal, review, publication, and presentation of the paper titled "Layout-Driven Scan Chain Partitioning and Reordering," and authored by Barbagallo et al. Magma is not aware of the present addresses or telephone numbers of these individuals.

79. Yervent Zorian, AT&T Bell Laboratories

80. Rabindra K. Roy, NEC USA

81. Michael Nicolaidis, TIMA

82. Mukund Modi, Naval Air Warfare Center

83. Kaushik Roy, Purdue University

84. Adit D. Singh, Auburn University

85. Sreejit Chakravarty, SUNY Buffalo

The subject matter for the foregoing individuals' information is: proposal, review, publication, and presentation of the paper titled "Scan Insertion Criteria for Low Design Impact," and authored by Barbagallo et al. Magma is not aware of the present addresses or telephone numbers of these individuals.

86. Gary Wong

87. Saeid Ghafouri

88. Rajeev Madhaven

89. Gregory C. Walker

90. Greg Wagenhoffer

91. Kam Kittrell

The subject matter for the foregoing individuals' information is: Magma's financial information and accounting, Magma's products and marketing, and/or Magma's business planning and forecasting. These individuals can be contacted through Magma's outside counsel, Pooley & Oliver LLP.

II. DOCUMENTS

Pursuant to Fed. R. Civ. P. 26(a)(1)(B), Magma identifies the following categories of document and data compilations it may use to support its claims or defenses. By providing these categories, Magma does not waive, and expressly preserves, any and all objections as to the relevance and admissibility of the documents.

1. Documents relating to Magma's electronic design automation products and technology.

2. Documents relating to trade secret source code for Magma's electronic design automation products.

3. Documents relating to prior art for U.S. Patent Nos. 6,434,733, 6,766,501, 6,192,508, including any related patents to the foregoing patents, and 6,505,328, 6,857,116, 6,854,093, 6,931,610, and 6,519,745.

4. Documents relating to prosecution of U.S. Patent Nos. 6,434,733, 6,766,501, 6,192,508, including any related patents to the foregoing patents, and 6,505,328, 6,857,116, 6,854,093, 6,931,610, and 6,519,745.

5. Documents relating to Synopsys's communications and representations regarding Magma, Magma's products, Magma's patents, Magma's technology, and Magma's pending lawsuits.

6. Documents relating to the areas of logic synthesis, testing, and/or physical design in the electronic design automation field.

All such documents are located at Magma's offices in Santa Clara, California or at the offices of its outside counsel, Pooley & Oliver LLP except as noted below.

Magma may also rely on documents disclosed by Synopsys and/or third parties in this action to support its claims and defenses. Magma also reserves the right to identify additional documents once formal discovery has been commenced, and at such time that relevant information may become available. Magma also reserves the right to later identify documents that may be relied upon by any expert witness.

III. COMPUTATION OF DAMAGES

Magma is presently unable to calculate its damages caused by Synopsys's conduct, and reserves its right to supplement this disclosure or otherwise make information known to Synopsys as this case proceeds.

IV. APPLICABLE INSURANCE AGREEMENTS

Magma has determined that it does not carry any insurance that could satisfy all or any part of a judgment which may be entered in this action.

Dated: February 28, 2007

FISH & RICHARDSON P.C.

By: _____

William J. Marsden, Jr. (#2247)
(marsden@fr.com)
919 N. Market Street
Suite 1100
P.O. Box 1114
Wilmington, DE 19899-1114
Telephone: (302) 652-5070
Facsimile: (302) 652-0607

Katherine Kelly Lutton *(pro hac)*
Tamara Fraizer *(pro hac)*
Joseph V. Colaianni, Jr. *(pro hac)*
Jennifer Ishimoto *(pro hac)*
FISH & RICHARDSON P.C.
500 Arguello Street, Suite 500
Redwood City, CA 94063
Telephone: (650) 830-5070
Facsimile: (650) 839-5071

Attorneys for Defendant and Counter-Claimant
MAGMA DESIGN AUTOMATION, INC.

Of COUNSEL:

James Pooley
L. Scott Oliver
Pooley & Oliver LLP
Five Palo Alto Square, 7th Floor
Palo Alto, CA 94306-2121
Telephone: (650) 739-7020
Facsimile: (650) 739-7060

CERTIFICATE OF SERVICE

I hereby certify that on February 28, 2007, I served the foregoing document on counsel of record in the manner indicated.

VIA ELECTRONIC MAIL AND U.S. MAIL

Karen Jacobs Louden
Morris, Nichols, Arsht & Tunnell
1201 North Market Street
Wilmington, DE 19801
Email: klouden@mnat.com

Attorneys for Plaintiff
Synopsys, Inc.

Valerie M. Wagner
Dechert LLP
1117 California Avenue
Palo Alto, CA 94304
Email: valerie.wagner@dechert.com

Attorneys for Plaintiff,
Synopsys, Inc.

Tamara Fraizer

50402348.doc